AD-A081 061 ENVIRONMENTAL RESEARCH AND TECHNOLOGY INC CONCORD MA F/G 17/9
AUTOMATIC WEATHER RADAR ECHO ASSESSMENT AND TRACKING.(U)
MAR 79 R K CRANE F19628-78-C-0076
UNCLASSIFIED AFGL-TR-79-0248 NL

1 OF 2
AD
A081061

AFGL-TR-79-0248

ADA081061

LEVEL II

AUTOMATIC WEATHER RADAR ECHO ASSESSMENT
AND TRACKING

Robert K. Crane

Environmental Research & Technology, Inc.
696 Virginia Road
Concord, Massachusetts 01742

March 1979

Final Report
15 February 1978 - 30 September 1978

DDC FILE COPY

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

80 2 25 015

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: AFGL-TR-79-0248	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): AUTOMATIC WEATHER RADAR ECHO ASSESSMENT AND TRACKING		5. TYPE OF REPORT & PERIOD COVERED: Final Report, 15 Feb 19[illegible] - 30 Sept 1978
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Robert K. Crane		8. CONTRACT OR GRANT NUMBER(s): F19628-78-C-0076
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Environmental Research & Technology, Inc. 696 Virginia Road Concord, Massachusetts 01742		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62101F 667007AA
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Geophysics Laboratory Hanscom AFB, Massachusetts 01731 Monitor/A. Chmela/LYW		12. REPORT DATE: March 1979
		13. NUMBER OF PAGES: 145
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report): Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

weather radar, echo areas, convective cells, severe storms, tangential shear, automatic cell detection, automatic cell tracking

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Previously developed algorithms for automatic radar cell detection and tracking were adapted for real-time use on the AFGL Echo Track and Significance Estimator. Additional significance estimation algorithms were developed to reduce the number of detected cells to a manageable number for display and interpretation.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

391776

TABLE OF CONTENTS

A | 23/ C.V.

ACKNOWLEDGMENTS

The results described in this report are the culmination of a sequence of contracts with the Air Force Geophysics Laboratory: "Parameterization of Weather Radar Data for Use in the Prediction of Storm Motion and Development", F19628-76-C-0264; "Development of Techniques for Short-Range Precipitation Forecasts", F19628-77-C-0058; and the current contract F19628-78-C-0076. Data used for the refinement of the algorithms were obtained by ERT under contracts with the Bureau of Reclamation, U.S. Department of the Interior, Contract No. 14-06-D-7673, and the Federal Aviation Administration, Amendment Agreement No. 4 to the Bureau of Reclamation contract.

Mr. A. Koscielny was the computer programmer operator for the Joint Agency Doppler Technology Tests in Norman, Oklahoma. Messieurs J. Leslie and G. Gustafson provided the programming support required to prepare the Interdata 7-32 computer programs.

1. INTRODUCTION

1.1 Program Objectives

The goal of the work reported herein is the real time operation of the cell detection and tracking algorithms previously developed by Environmental Research & Technology, Inc. (ERT) for the Air Force Geophysics Laboratory (AFGL). Specifically, the work included the following tasks: (1) encoding the automatic echo assessment and forecasting algorithms developed under Contract No. F19628-77-C-0058 on the Echo Track and Significance Estimator (ETSE) Interdata Model 7-32 Computer; (2) testing the algorithms for real time operation; (3) operating the computer and writing assembly level programs on the Interdata 7-32 computer during the Joint Agency Doppler Technology tests at the National Severe Storms Laboratory, Norman, Oklahoma, March through June 1978; and (4) refining the automatic assessment and forecasting algorithms based upon a critical radar meteorological analysis of the use of the algorithms.

The cell detection and tracking algorithms were developed to automatically process weather radar data to provide real time identification of severe weather and short range (0-20 minute) forecast of regions of potential hazard to aircraft operation.

1.2 Summary of Results

The cell detection and tracking algorithms previously developed under Air Force Contract (Crane, 1978; Crane, 1977) were designed for real time operation on a medium scale computer based on our experience with an extremely limited data sample (five consecutive azimuth scans) and with the CDC 6600 computation facility at AFGL. The subsequent development of a viable real time weather radar data processing system for use by the Air Force, however, required access to a significantly larger set of weather radar data.

The initial development of the cell detection technique had been undertaken for the Federal Aviation Administration (FAA) (Crane, 1976) for use in automatic air traffic hazard detection. Their continued interest in the processing scheme and its evaluation resulted in a contract between ERT and the FAA to process several hours of radar observations made simultaneously with aircraft penetrations. ERT was also under

contract with the Bureau of Reclamation, Department of Interior (BuRec) during the period of performance of this contract for the acquisition and analysis of significant amounts of radar data for the development of precipitation augmentation strategies for the High Plains (HIPLEX). As a part of that contract, the cell detection and tracking algorithms were installed on the CYBER-74 Computer System in Denver and used to obtain statistical data on the spatial organization of precipitation production within high plains storms.

The three concurrent programs, the development of real time techniques for the Air Force, hazard detection algorithm evaluation for the FAA, and the spatial organization analysis for BuRec provided the experience with a significantly larger data base needed for the refinement of the tracking algorithms (Task 4) and the development of a viable real time processing program.

The work under this contract was organized in the four tasks listed in Section 1.1: (1) encoding the algorithms for real time operation on the Interdata 7-32 computer; (2) testing the algorithms for real time operation; (3) operating the ETSE in Norman, Oklahoma during the Joint Agency Doppler Technology tests; and (4) refining the algorithms for improved tracking and short range forecasting. ERT provided a programmer operator for the 1978 Joint Agency Doppler Technology tests in fulfillment of Task 3. The detection and tracking algorithms refined as a result of Task 4 have been coded and installed on the Interdata 7-32 computer in fulfillment of Task 1. Their description is the subject of this report. Program listings and a copy of the operating instructions are included in Appendix C and D. The algorithms have been tested in compliance with Task 2 and, as coded and operated, performed in real time.

The algorithms operate in real-time on the Interdata 7-32 computer as required. Real-time operation on the Interdata 7-32 in the manner used in non-real-time analysis, however, requires additional programs to fetch and store the raw radar data. For real-time operation at the level of performance of the previously developed non-real-time program, a new operating system is required for the Interdata 7-32 computer which utilizes the real-time interrupt capability of the machine to run the cell detection and tracking programs as background programs with the data averaging and storage programs in foreground. Such an undertaking was

beyond the scope of this contract, and a simpler program has been specified which allows operation in real-time on alternate scans; one to fetch and store the data, the second to process the data. This program, a modification to the TSE program provided by Raytheon (Boak et al, 1977), is listed in Appendix C.

1.3 Software Development

The previously developed cell detection and tracking programs were extensively modified for use on the Interdata 7-32 computer. The tracking program described in the final report of the previous contract (Crane, 1978) was completely rewritten starting with a new set of algorithms. Experience with the larger volume of radar data available from BuRec forced the program revision. Two major problems existed with the initial tracking program; excessive computer storage requirements for the large numbers of cells encountered in practice and an inherent inability of the program logic to separately establish individual smoothed track velocities for each cell. The new tracking program develops the volume cell attributes discussed in the previous contract report, establishes the existence of cell clusters, provides an estimate of cell significance, and maintains both instantaneous and smoothed velocity estimates for each cell.

The cell detection program encoded for use on the Interdata 7-32 computer is a streamlined version of the original cell detection program. The fixed contour outlines are developed not as line segments enclosing the contoured area but as azimuth strobes within the echo region in conformity with the ETSE display scheme. Attributes are not generated for the fixed contours. The cell detection subroutine operates as before. Addressing in the arrays used in the subroutine has been extensively altered to increase operating speed.

The programs can operate over a wide range of reflectivity thresholds but should be used at a relatively high reflectivity threshold, processing only data with reflectivity values above, say, 40 dBZ. This threshold was selected to reduce the number of cells being processed in the computer. The reduction in the number of cells being processed improves operating speed and matches the output requirements. Experience with storms in the high plains indicates that severe storms produce large

numbers of cells. Cell counts for a 20 dBZ reflectivity threshold are over 150 during the active period of a storm; the total number of separate cells observed during a storm often exceeds several thousand. By way of contrast, the output requirements established by the remote display system (communicated by AFGL personnel), are for not more than 12 cells at any one time. The more than an order of magnitude reduction in active cell count can best be accomplished by increasing the threshold reflectivity value.

In addition to reflectivity threshold selection, significance is established using the integrated tangential shear of the radial velocity for each cell. Tangential shear cell detection as previously coded (Crane, 1977) is not attempted in the streamlined version of the program. The tangential shear data are used to develop a shear attribute for each cell detected on the basis of reflectivity alone. The program operates by detecting all cells that occur above the processing threshold but saving for tracking and output only the 16 cells having the highest reflectivity, integrated tangential shear product. Internally, the program processes 31 volume cells but only the 12 volume cells with the highest reflectivity, shear product are output after each volume scan. The program can be modified to process and output data for more cells by changing array sizes and test limits.

The object of the work reported in this contract was to streamline the original version of the cell detection and tracking programs for real-time use on the Interdata 7-32 computer with the operating system provided by Interdata. Many of the features of the original program, such as the generation of fixed contour attributes and the independent detection of tangential shear peaks, were removed to establish the real-time program. These features may be recovered only if the original version of the program, the program operating on the AFGL CDC-6600 computer, is installed on the Interdata 7-32 for non-real-time processing.

1.4 Organization of the Report

This report considers only the software developed for use on the FTSF, documentation for Tasks 1, 2 and 4. Task 3 covered the programmer computer operator for the 1978 measurements in Norman, Oklahoma. The results of that task were reported in the quarterly reports and will not be considered further.

Background material and algorithm refinement based upon results from the FAA and BuRec programs are considered in Section 2. Section 3 documents the program for the ETSF. Section 4 summarizes program status and makes recommendations for future work. Program listings, flow charts, variable definition, and operating instructions will be found in the appendices.

2. BACKGROUND

2.1 Overview of Automatic Processing Scheme

Conventional weather radars produce large amounts of data - a significant fraction of which is highly redundant. Doppler radars produce even larger amounts of data. Significant weather events may be imbedded in the mass of redundant data. It is the goal of the automatic processing scheme to extract the relevant information from the mass of data to (1) reduce the data transmission requirements for the communication of weather data obtained from a radar, (2) to screen the data prior to display to meteorologists, (3) to preprocess the data for automatic hazard detection, and (4) to prepare the data for use in objective short range forecasting.

The processing scheme is structured to use the cell detection algorithm in on-radar-site computers to perform the bulk of the data reduction. The cell data are then communicated to regional computers (or to a second program in a stand alone radar data processor) for tracking and interpretation. For a national network of weather radars, the tracking program would accomplish the task of netting different radars and developing a single, best estimate description of the current weather for use in displays, hazard detection and warning, and short range forecast.

For this contract the cell detection and tracking algorithms are operated in a single computer; the final output is track data for the 12 most significant cells. The track data include smoothed cell velocities which are used in the tracking program for data association and may be used externally for short range forecast.

The output cell and track data are for significant features in the larger mass of radar observations. Significance is defined in an ad hoc manner using cell intensity, area, vertical development, and tangential shear data. Parameters that are intuitively associated with significant events such as severe hail, severe thunderstorm turbulence, and tornadoes have been selected for the determination of a significant cell. Operational experience with the processing algorithms and a large sample of data is required before the values of the thresholds used to establish cell significance can be refined. The current algorithm has been

partially tested using aircraft penetration data. For the measurements currently available, a positive correlation has been obtained between the location of significant cells and aircraft turbulence. An example of this association is presented in Figures 1 and 2 (output obtained from the work at ERT sponsored by the FAA). The significant cells are indicated by the tightly clustered symbols for reflectivity values greater than 40 dBZ. The time marks are 6 km apart along the aircraft track. The aircraft was within a typical cell diameter of two significant cells between 1640 and 1641, a time marked by the strongest acceleration fluctuations (turbulence) during the penetration. By way of contrast, the remainder of the penetration was quiet and did not show strong acceleration fluctuations and was not in the vicinity of significant cells.

The display in Figure 1 immediately identifies the locations of the significant cells and graphically presents the essential data contained in the radar observations. In contrast, a section of a conventional contour display and of the cell display are presented in Figure 3. The essence of the data is immediately evident in the cell display. The important 45 dBZ cell that results from the strong updraft depicted between 1640 and 1641 MDT in the aircraft data is observed in the cell display but not in the contour display. This cell produced the strongest turbulence. Even with a color display, this important region would not be evident although the higher reflectivity contours in Figure 3a would stand out more vividly, in the manner of the darkened symbols in Figure 3b.

A display such as Figure 3b is readily interpreted but is not the normal weather radar display familiar to trained radar meteorologists. The cell display provides the important details but at a scale that is smaller than used by most meteorologists and since many cells may be observed at any single time, a display of all the cells may prove to be confusing. The number of cells active at any one time with reflectivities above 15 dBZ and the number of significant cells for a set of observations of showers in Kansas (output from data processed by ERT for BuRec) are depicted in Figure 4 as a function of time and volume scan sequence. The total number of cells present within a 25 to 150 km annulus of Goodland, Kansas approached 200 during the most active part

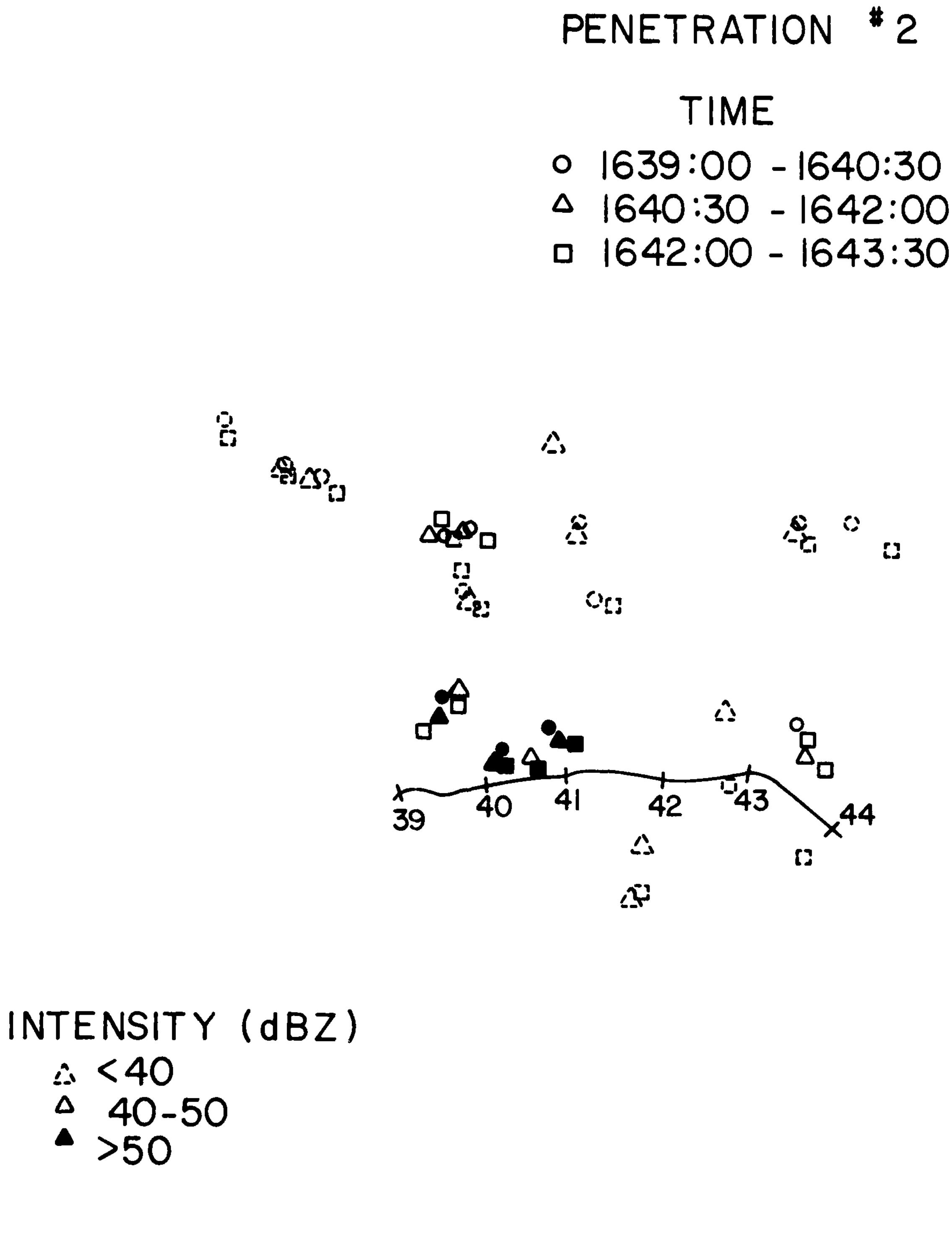


Figure 1 Aircraft Penetration 22 July 1976 as Observed by the Grover Radar

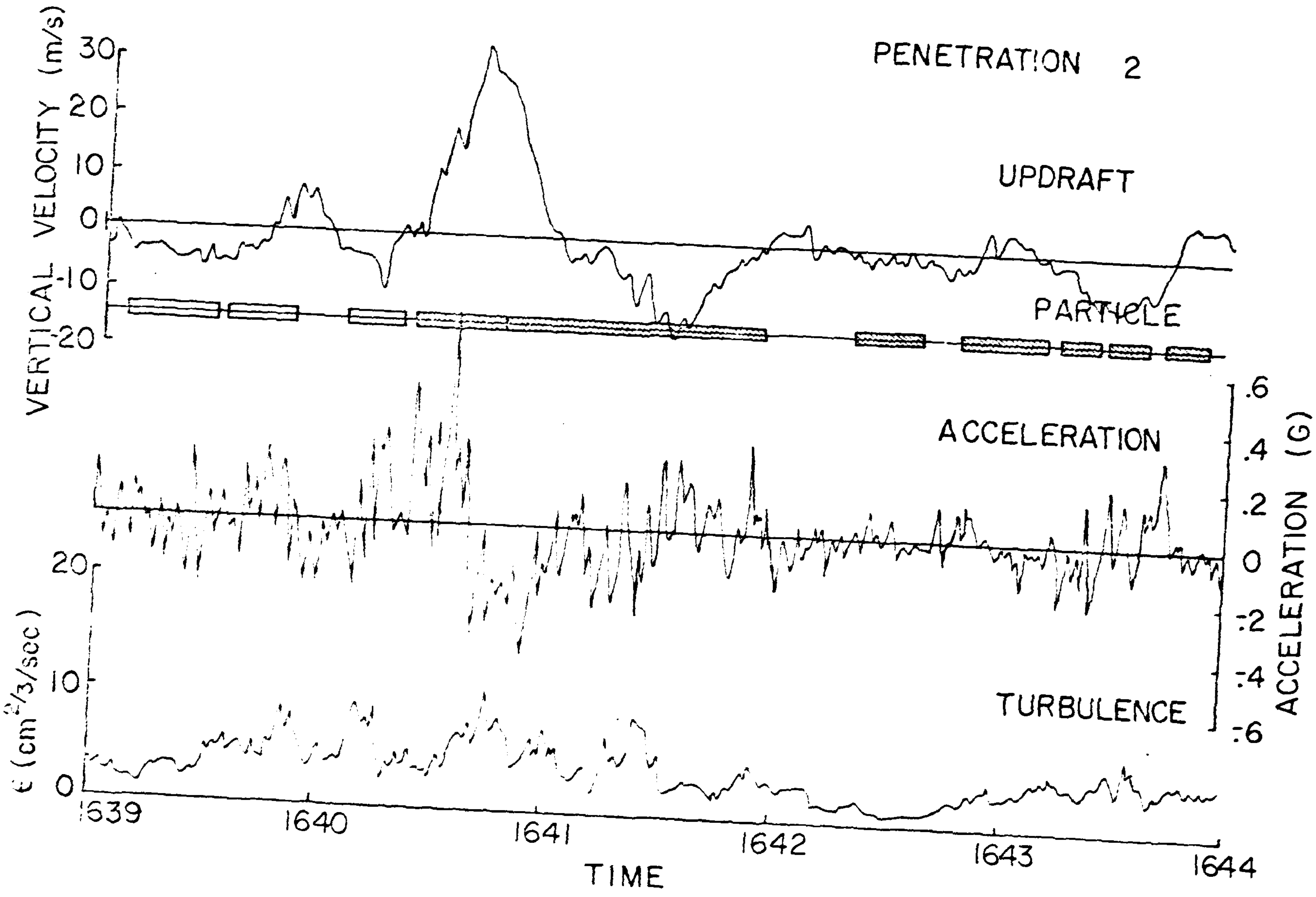


Figure 2 Aircraft Observations During Storm Penetration

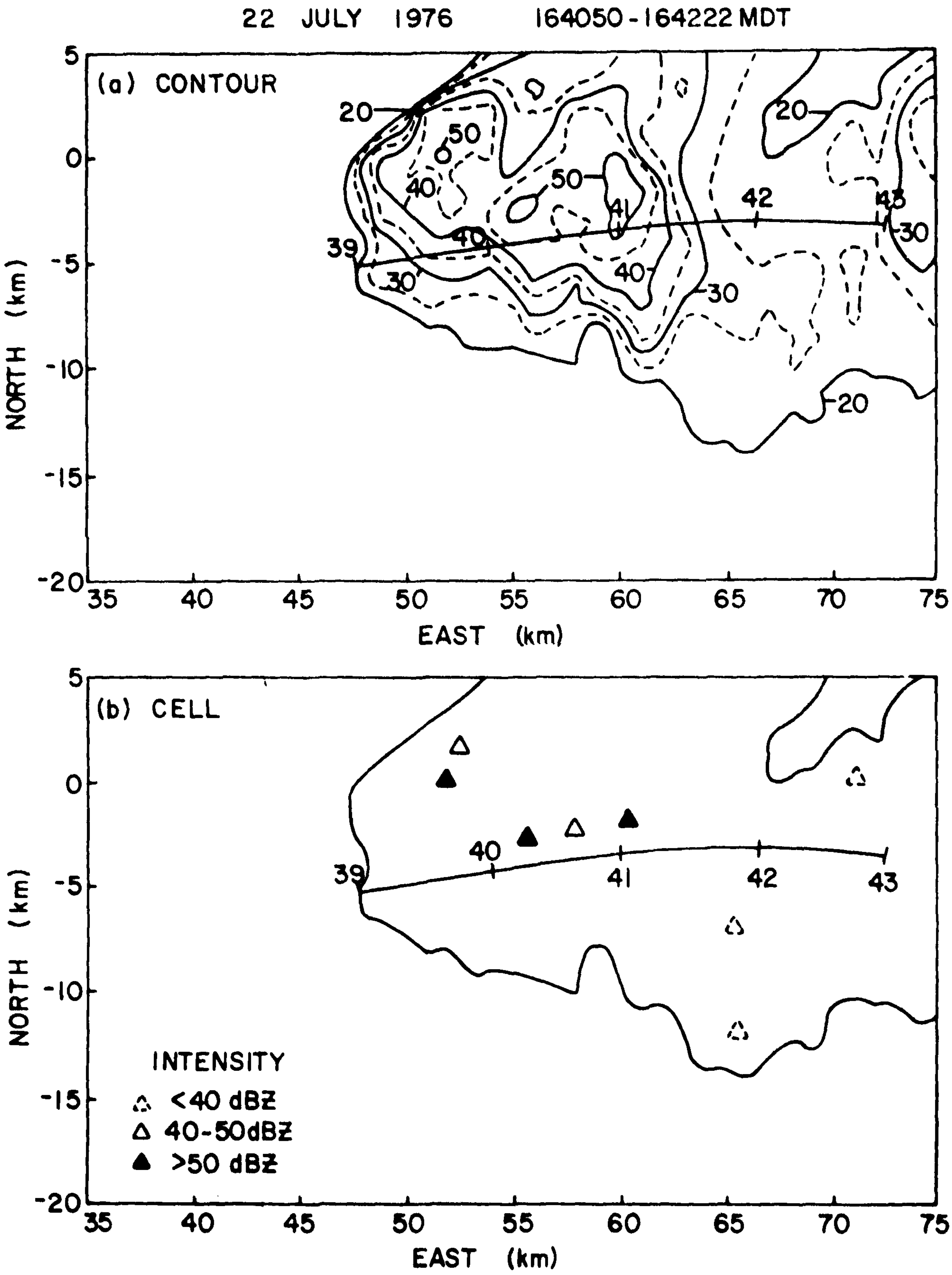
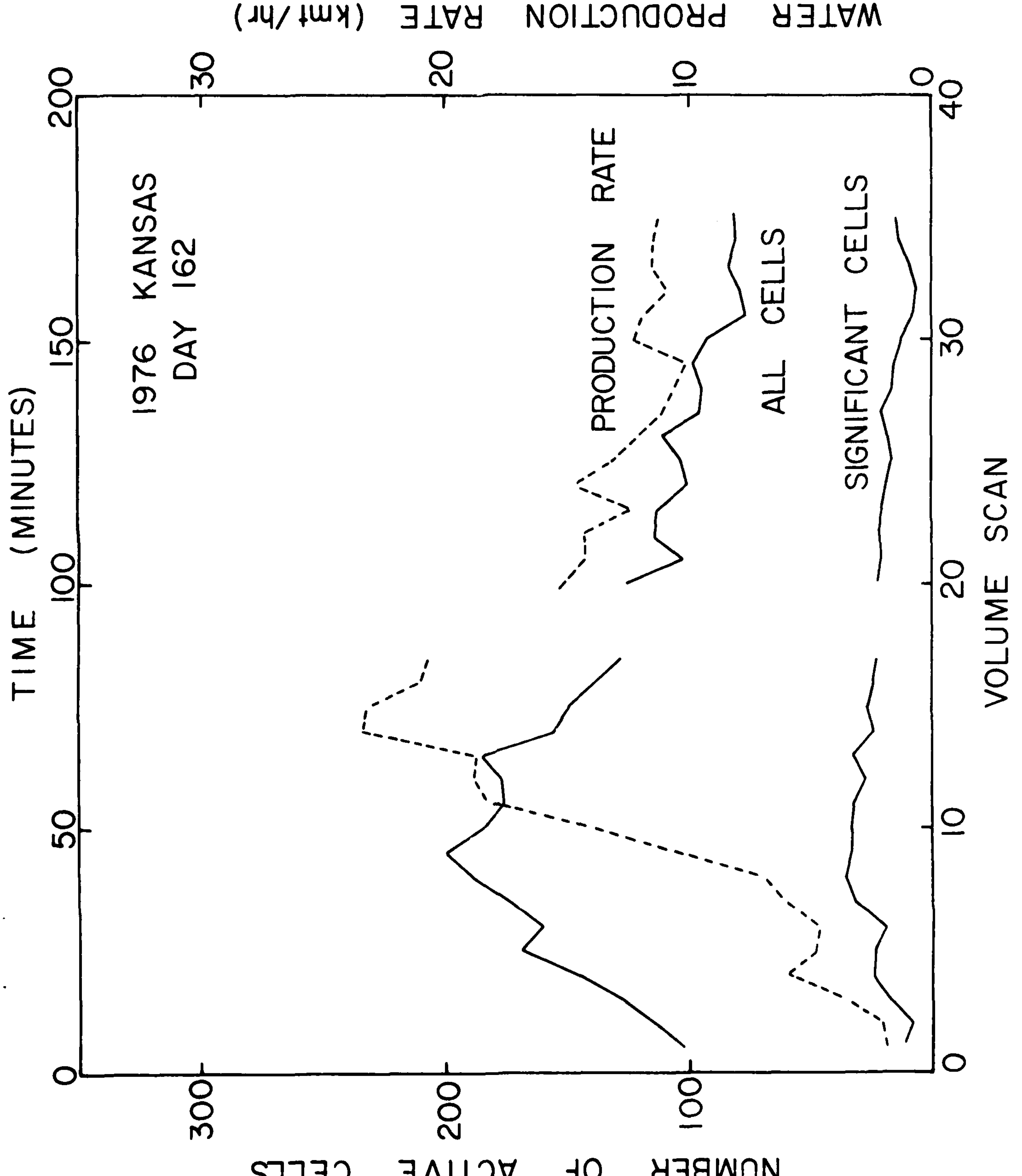


Figure 3 Contour and Cell Data, Grover Radar Data

Figure 4   Active Cells and Total Water Production, Goodland, Kansas
10 June 1976

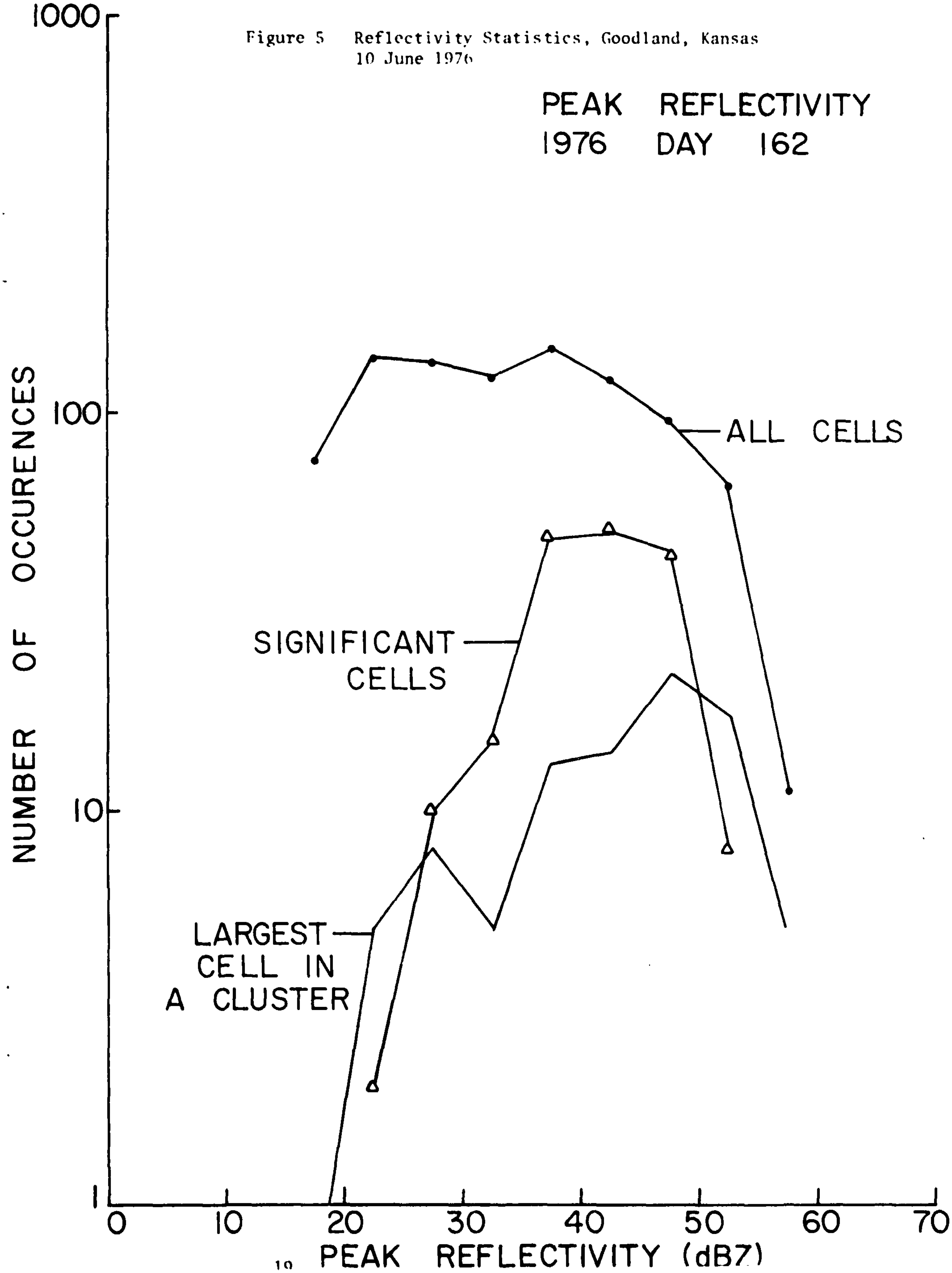


of the storm. The number of significant cells was as high as 36. A display of all the cells would be very difficult to interpret. A display of only the significant cells, though readily interpreted by a computer, may still be difficult for a meteorologist to interpet. A further reduction in the number of significant cells is required. The program developed under this contract utilizes tangential shear data in addition to reflectivity information to further reduce the number of cells for display as the most significant cells. The utility of this algorithm for significant cell selection still needs verification.

The small cells detected and tracked by the algorithms developed under this and prior contracts are well behaved in time and space. The cells show vertical development, persist, and have average velocities that approximate a steering level wind. Summary statistics for the June 10, 1976 storm displayed in Figure 4 are presented in Figures 5 to 9. These data provide statistical summaries of several cell characteristics representing either the values averaged over the lifetime of a cell (average) or the peak value obtained during the cell lifetime (peak). Lifetime data are presented in Figure 10. The data in Figure 5 depict the statistics of the highest reflectivity values reached by a cell during its lifetime. The data are drawn from a sample of 900 cells whose lifetimes exceeded 10 minutes. These data show that reflectivity alone was not used as a criterion for significance. Over 77 cells had a peak reflectivity in excess of 50 dBZ while only 8 significant cells had a peak reflectivity in excess of 50 dBZ. In the processing used to obtain these data, significance was defined based upon the vertical structure and horizontal dimension of the cell as well as its reflectivity. A high reflectivity echo that was observed only at one elevation angle was not considered to be a significant cell.

The cells detected by the processing scheme have relatively small areas. The average cell area for all cells and for significant cells do not differ as indicated in Figures 6 and 7. The peak cell diameter is of the order of 3 km. The cells tend to be vertical structures. Statistically, the area of a cell at the lowest elevation angle does not differ from the area of the cell at the height at which it has a maximum reflectivity value. Note that the area at any height is defined by a contoured region 3 dB below its peak reflectivity value at that height. At another

Figure 5 Reflectivity Statistics, Goodland, Kansas
10 June 1976



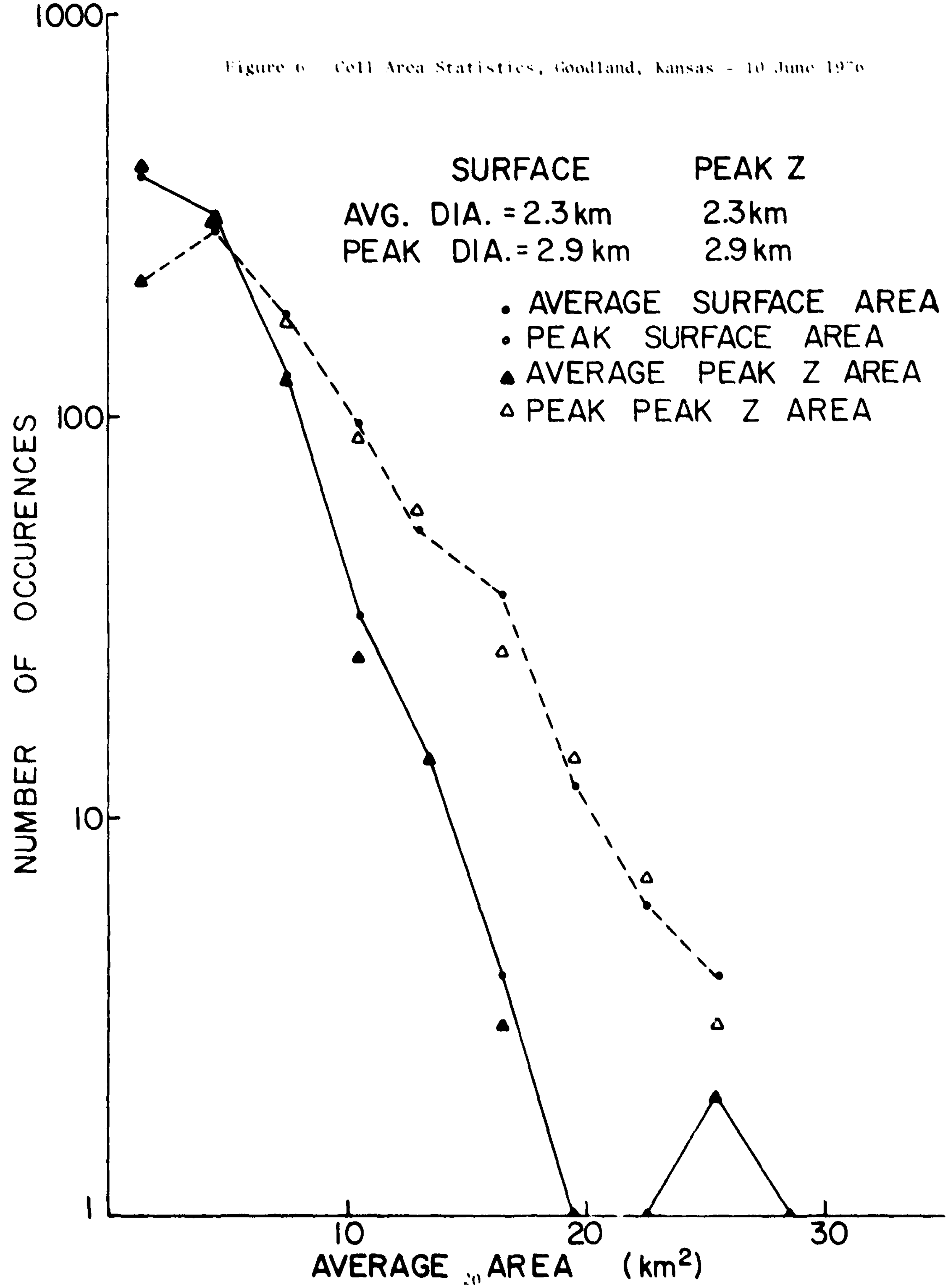


Figure 6 Cell Area Statistics, Goodland, Kansas - 10 June 1976

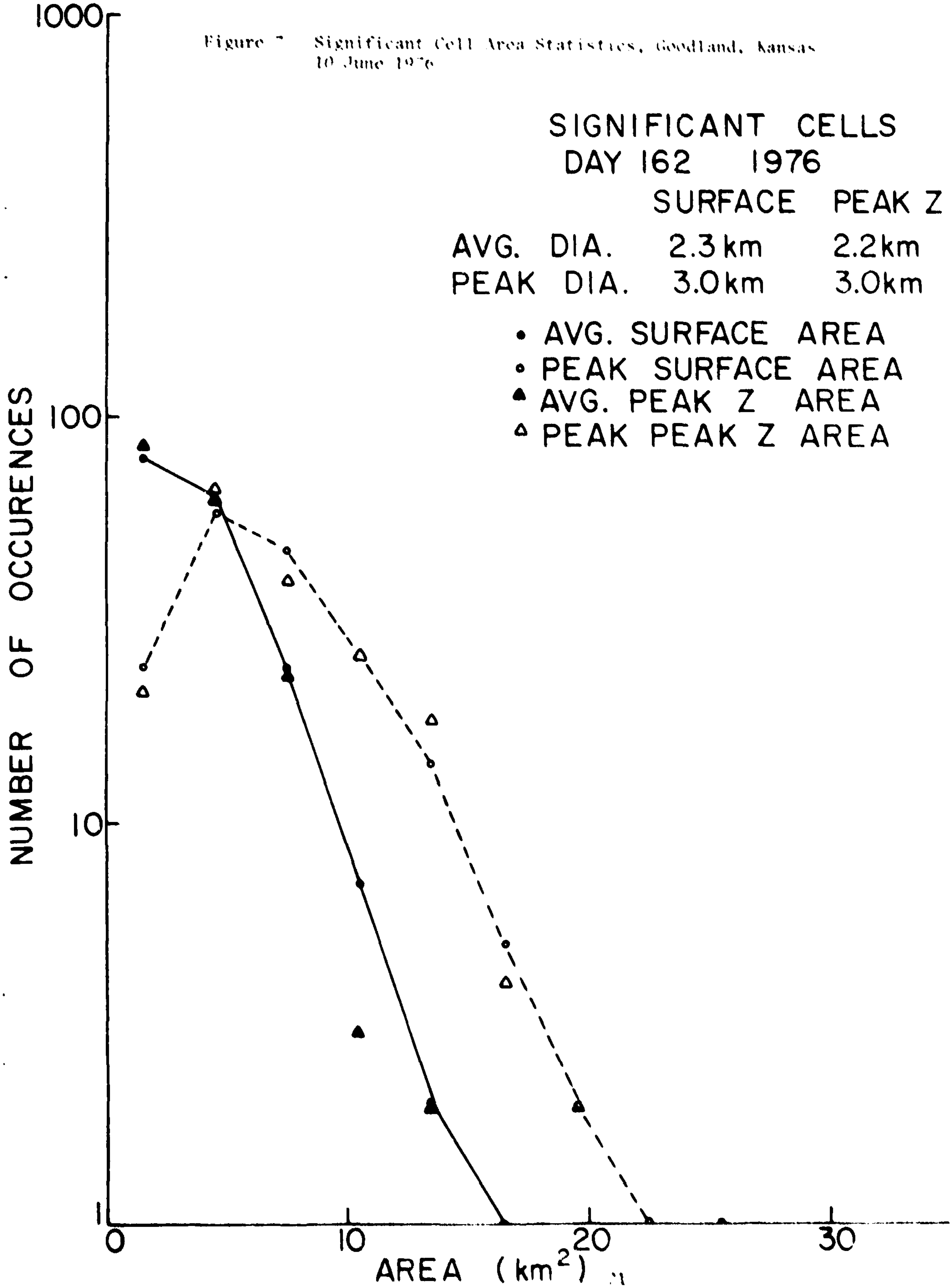


Figure 7 Significant Cell Area Statistics, Goodland, Kansas
10 June 1976

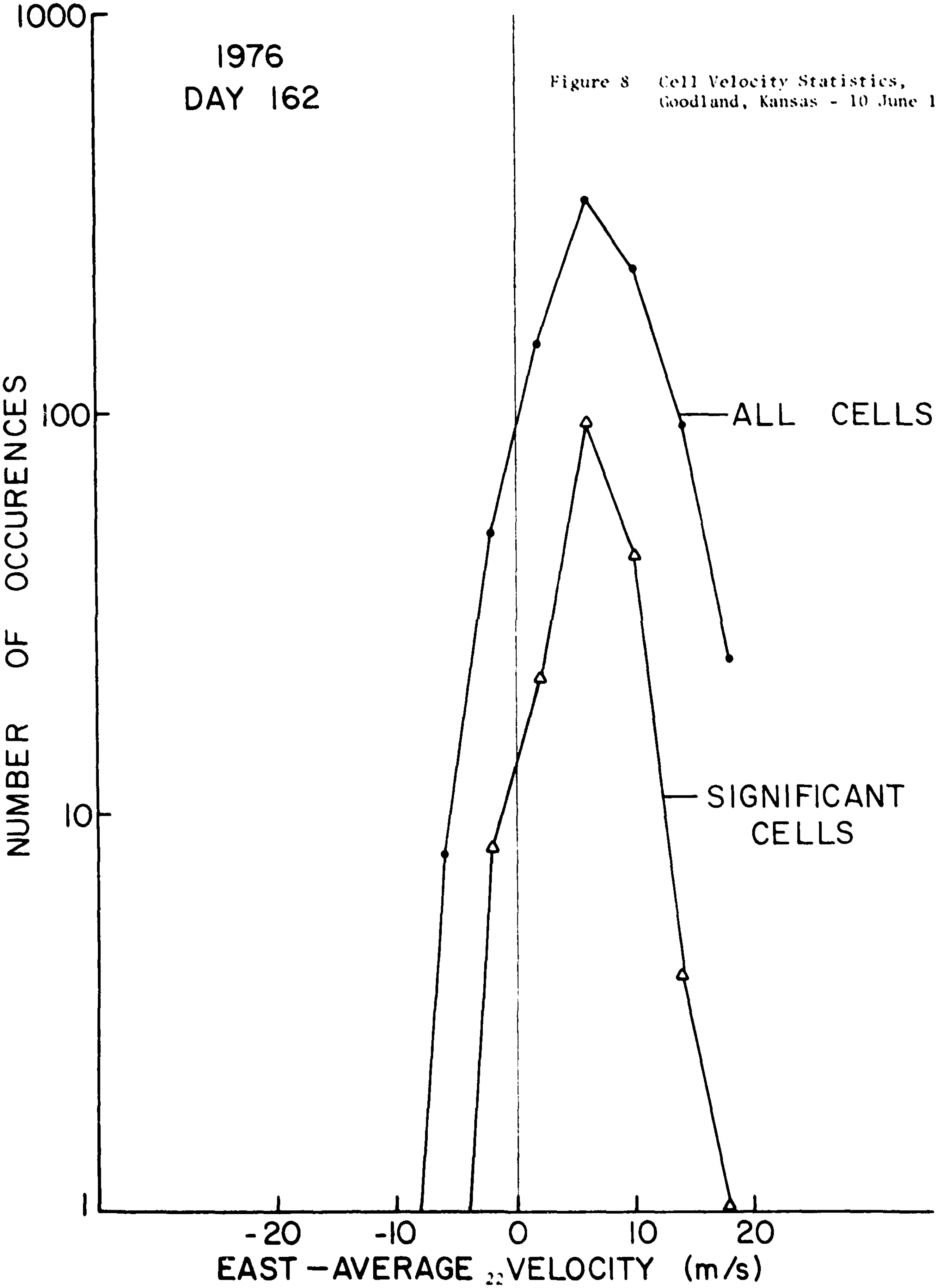


Figure 8 Cell Velocity Statistics, Goodland, Kansas - 10 June 1976

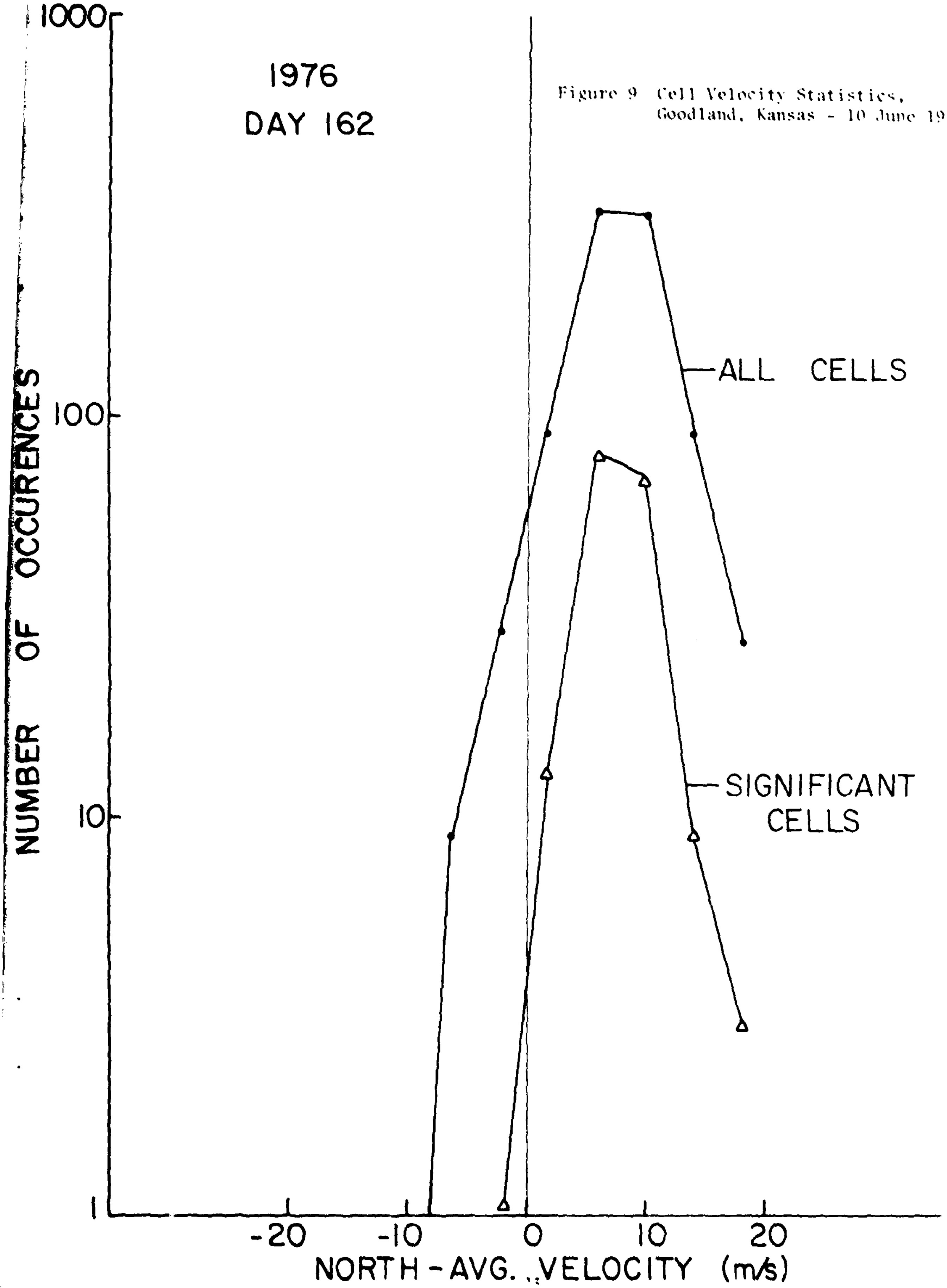


Figure 9 Cell Velocity Statistics, Goodland, Kansas - 10 June 1976

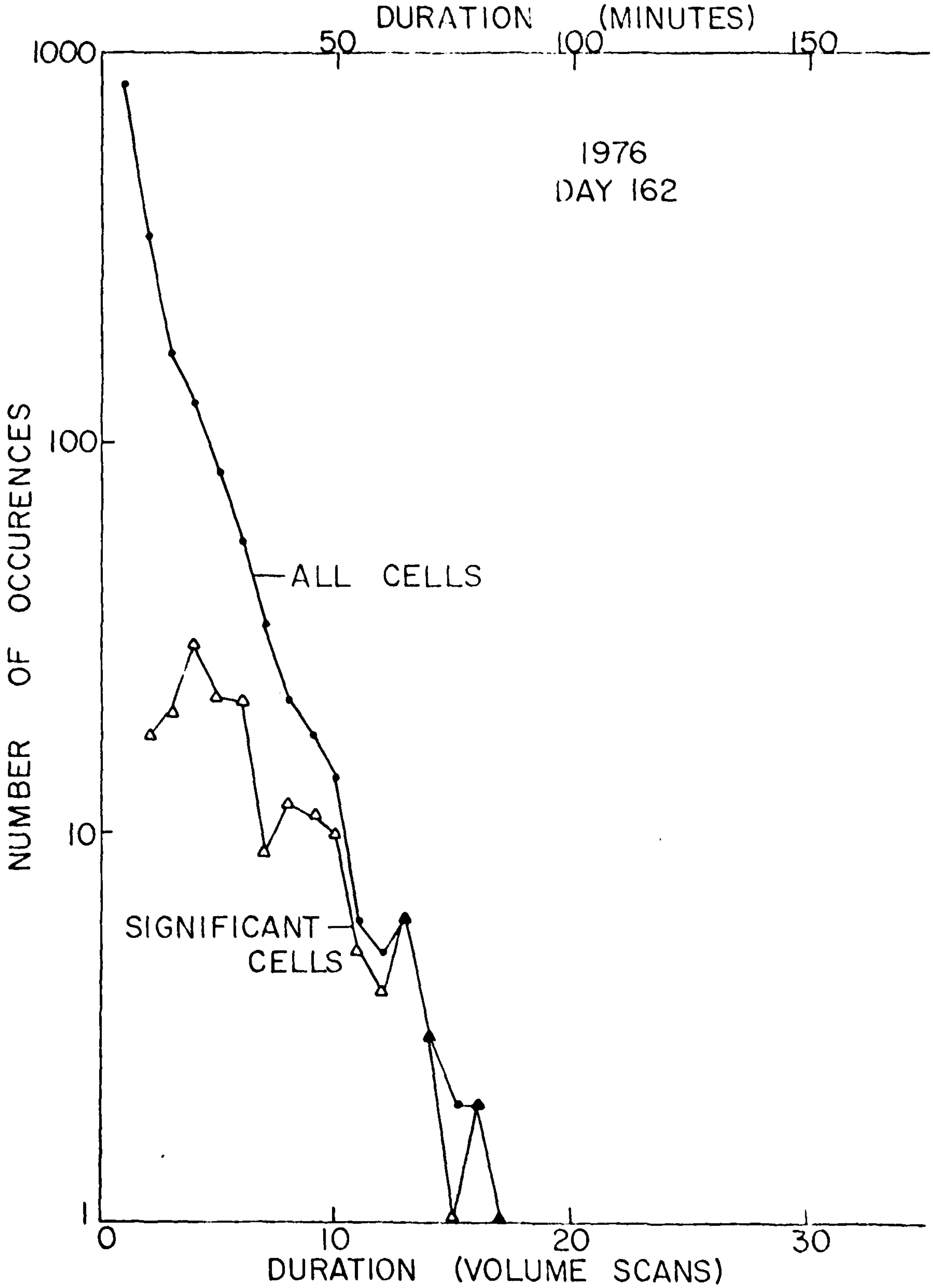


Figure 10 Cell Lifetime Statistics, Goodland, Kansas - 10 June 1976

height, the contour used to define its area will have a different reflectivity value.

The tracking algorithms track each cell individually. The distributions of cell velocity averaged over the cell lifetime are depicted in Figures 8 and 9. These data show no significant differences between the velocities of all the cells and of significant cells. The individual track velocities may differ from the mean (or steering level wind) velocity by as much as 4 m/s (rms in each component of the wind). Observations show that the deviations are not entirely random. Larger scale convergence and divergence patterns are evident in the cell trajectories. On a smaller scale, it is evident that cells affect each others motion. A tendency has been observed for cells to follow each other along the same track even though they may not have initially developed along the track.

The cells persist for a range of lifetimes. On average, cells with reflectivities above 15 dB last for a little over 12 minutes. Significant cells last over 30 minutes on average. These results show that ideally, a full cycle of radar observations (a volume scan) should be acquired in between four and five minutes to get more than two observations of a cell during its lifetime. Practically, a longer time, six to seven minutes, is required for the processing algorithms as implemented under this contract. Processing speed can be increased by preparing a new operating system for the Interdata 7-32 computer but this was beyond the scope of the contract. The motion of significant cells, if detected early during its development, may be extrapolated for upwards of 20 minutes before they disappear indicating that short range forecasting of cell location is feasible.

2.2 Cell Detection

The cell detection algorithms have been previously defined (Crane, 1977, 1978) and will not be detailed again. Flow charts for the processing algorithm are presented in Appendix B. Briefly, a cell is a region within a contour, a fixed number of quantization steps below a local maxima that includes no other cells. For most observations, a quantization step of 1 dB and the use of contours 3 dB below a local reflectivity maxima seems to work best. The quantization step and contour threshold

were empirically established by Crane (1976) using data from Wallops Island, Virginia. The 1 dB step and 3 dB threshold produced a detection probability better than 0.95 on these scans and a false alarm rate of less than 2 per scan. By increasing the threshold, the false alarm rate was reduced but at the expense of a lowered detection probability.

Cell detection is performed for all localized reflectivity maxima that exceed a processing threshold. In the post mission processing versions of the cell detection program, attributes are obtained for all the cells detected above the lowest threshold fixed reflectivity processing contour (fixed contour or echo region). The streamlined edition of the cell detection program provides output only for the most significant cells.

2.3 Tracking

An entirely new tracking algorithm was developed during the period covered by this contract primarily for application to the post mission processing requirements of the FAA and BuRec contracts. This program was subsequently modified for real time application under this contract. The real time version generates volume cells and volume cell tracks.

The same tracking algorithm is used to generate the 3-dimensional cell from the successive azimuth scans within a volume scan sequence and to track the volume cell in time. Each new volume cell is identified using cell data from a single azimuth scan that cannot be associated with other, previously established tracks. Association is attempted with the newly detected volume cell on each successive azimuth scan until the track is terminated. A track is terminated when no new data are obtained for a volume cell during a complete volume scan cycle.

Association is established using the location of the cell on a azimuth scan as compared with the volume cell location extrapolated to the time of the azimuth scan together with the differences in the heights of the last observed data and the current data, differences in cell areas, and differences in reflectivites. A measure of the goodness of an association between a cell and track, is established for each possible track, cell combination. The final cell track pairings are those that minimize the sum of the measures (maximize the goodness of association) for all the cells and tracks that may be possibly paired during the association

process. The set of cells and tracks that may possibly be paired are defined as a cluster.

The volume cell track is the primary entity maintained by the tracking program and successive radar observations are associated with the expected position of the cell along its track at the time of the radar observation. The tracking algorithm was developed in this manner to allow the use of data from more than one radar system since all that is needed for the association algorithm is the location and time of the cell centroid as reported by each radar together with other attributes such as reflectivity, tangential shear, etc.

Two velocities are maintained for each cell track - an instantaneous velocity, the difference in cell centroid location divided by the time interval between observations, and a smoothed velocity obtained by low pass digital filtering of the raw velocity data. The coefficients in the filter were established by trial and error using a large data sample. The initialization of the track velocity for each cell is important due to the extrapolation process used for tracking. Experiments with the tracking program show that adequate results are obtained if a zero velocity is used for the initial velocity estimate but better results can be obtained if an estimate of the steering level wind is used for the initial tracking velocity. The program automatically updates the initial velocity estimate after processing sufficient data to establish a stable estimate. The measure of success of the tracking program was taken as the rate required for the smoothed track velocities to stabilize.

The cell detection and tracking programs were initially developed to process a large number of cells, up to 512 active cell tracks at any one time and to calculate upwards of 30 attributes for each cell, cluster, and fixed contour (echo) region. Processing this amount of data is not possible in a real time environment with storage and display limitations. The basic algorithms for cell detection and tracking have been maintained. The number of tracks to be processed has been reduced by increasing the reflectivity threshold for processing and by incorporating the tangential shear information in the decision process for saving the most important 12 to 16 cells of 30 or more cells that exceed the reflectivity processing threshold. Further storage savings have been accomplished by reducing the number of attributes for each cell.

The cell detection and tracking programs were not coded in efficient manner for operation on the ETSE Interdata 7-32 computer. Extensive program revision was performed to reduce the number of subroutine calls and to revise the addressing procedure to reduce the time required to fetch or store a variable. The result is a streamlined cell detection and tracking program that will handle a reasonable number of active cells during the time required for an azimuth scan. Specifically, over 100 cells can be detected and processed in less than 52 seconds using the Interdata 7-32 programs. The processing time can be reduced further by dynamically varying the reflectivity processing threshold to maintain fewer than say 20 detected cells but this has not been necessary.

The programs to fetch the raw radar data and prepare the data for use in the cell detection program are included in Appendix C and D. The real program is designed to permit data gathering and cell detection on alternate scans. Real-time processing using the cell detection algorithms on every azimuth scan is possible with the Interdata 7-32 computer but will not occur unless considerable effort is expended to develop a new operating system for the computer tailored to use the interrupt and background/foreground processing capabilities of the computer to provide quasi-real-time cell detection and simultaneous real-time radar data acquisition, averaging and storage. Sufficient time is available for all the programs to operate on all the data from a volume scan within the time of a volume scan but the processing of the data from the lower elevation angles will lag behind data acquisition and only catch up on the higher elevation angles.

3. PROGRAMS FOR THE AFGL TRACKING AND SIGNIFICANCE ESTIMATOR

3.1 Processing Options

Two programs exist for operation on the TSE. A post mission processing program is available called CRANI that reads previously prepared data from the disk and performs cell detection and tracking. This program is not intended for real time operation and threshold levels may be reduced to allow detection of a large number of cells. The program is intended for post mission data analysis when time is not at a premium and larger amounts of output can be handled by the user. The input data must be prepared for storage on the disk using a modification to the TSE programs generated by Raytheon. The program is called ERT and is listed in Appendix D.

The operational program uses the same cell detection and tracking subroutines but is called from a modified version of the Raytheon provided TSE programs. These programs store the data on one azimuth scan and process the data on the next. Data from the first, third, and fifth scans are processed during the second, fourth, and sixth. During the seventh scan the displays are prepared and the programs are reinitialized for the next cycle of the seven scan sequence.

All the programs generated under this contract for use on the Interdata 7-32 computer are listed in Appendix C and D. CRANI is the post mission main program which calls CONTOR and TRACK. The output from this program is stored on disk for subsequent listing. This program requires the use of a preprocessing program PPRDSC (modified for ERD) written by AFGL personnel and listed in Appendix D. The real time program includes modified versions of the TSE programs supplied by Raytheon. The modified programs are TSEMAIN and the subroutines RMAP and PPRDSC. The CONTOR and TRACK subroutines are called from RMAP. These programs are presented in Appendix C.

3.2 Cell Detection

The cell detection subroutines CONTOR and PEAKP were substantially modified to reduce processing time spent in addressing the data arrays. The principal modification was to change all the arrays to single dimen

sional arrays and to explicitly perform the address calculations in the program. In this way, multiple references to the same array location would not involve the time consuming recalculation of the address for each array reference.

CONTOR was extensively modified to remove the fixed contour attribute generation algorithm. Contour data are still prepared but in the azimuth strobe format used by the TSF display program. No radial-to-radial association is required for this streamlined version of the program significantly decreasing the length and complexity of the program.

PEAKD has been changed only to accept the modified system of addressing and to select only the 16 most significant cells for further processing. Reference to the fixed contour identification tag was also removed from PEAKD since the tags were produced in the association, attribute generation logic of the CONTOR subroutine which was removed for the Interdata version of the program.

5.3 Tracking

The new tracking program consists of the subroutine TRACK which calls COMPAR to perform the cell to track associations. The subroutine COMPAR searches the track list and the cell list from the last scan, and finds all possible pairs for which the goodness of fit measure does not exceed a preselected threshold. The measure is given by

$$M = 1. + |Z_C - Z_T| \cdot W_Z + |A_C - A_T| \cdot W_A + |H_C - H_T| \cdot W_H$$

$$+ [(X_C - X_T)^2 + (Y_C - Y_T)^2] \cdot W_L$$

where M is the measure, the subscripts C refer to cell and T to track and Z is reflectivity, A area, H height and X,Y are centroid locations. The weights W were set by trial and error. The current values are listed in Table 1. The best cell, track pairing has the lowest measure; pairings with a measure greater than M_p are not allowed.

Several pairings are possible in a cluster of cells. Subroutine RESOLV selects the best set of cell, track pairs in a cluster. The attributes are updated in ATRAK which is called from COMPAR if there is no cell cluster or from RESOLV if a cluster exists. The subroutine BTRAK

is also called to store cell data in the VR array each time a track is updated. This information is used in calculating the measure M. The VR array data are either from the lowest elevation angle on which the track was observed or from the last elevation angle at which it was observed. The measure used to evaluate the cell, track pairing is the minimum measure obtained using either the last or lowest elevation angle data.

At the end of a volume scan cycle, STRAK is called to calculate the attributes and to output the track data. Only 12 tracks are output from STRAK in the operational version although a maximum of 32 tracks are maintained at any one time. The list of the 16 attributes maintained for each track is given in Table 2.

TABLE 1

TRACKING, WEIGHTS, MAXIMUM MEASURE AND VELOCITY FILTER

W_X	-	1/10	$(dBZ)^{-1}$
W_A	-	1/25	$(km)^{-2}$
W_H	-	1/2	$(km)^{-1}$
W_L	-	1/5	$(km)^{-2}$
M_P	-	5.	

Velocity Filter:

$$(V)_N = a \frac{\Delta y}{\Delta t} + b\,(V)_N + c\,\bar{V}_N$$

$$(V)_E = a \frac{\Delta x}{\Delta t} + b\,(V)_E + c\,\bar{V}_E$$

V velocity; N,E refer to Northward and Eastward (y,x) components

$\Delta x, \Delta y$ = change in position between volume scans

Δt = time between scans

$\bar{V}$ = average velocity, all cells

a = .4 b = .3 c = .3

TABLE 2

TRACK ATTRIBUTES

1.	Observation Time	(seconds from start of year)
2.	East Location	(km)
3.	North Location	(km)
4.	Average Reflectivity	(dBZ)
5.	Volume	(km^3)
6.	Peak Reflectivity	(dBZ)
7.	Height of Reflectivity Peak	(km)
8.	Reflectivity at Lowest Elevation Angle	(dBZ)
9.	Area of Cell on Lowest Elevation Angle	(km^2)
10.	Height of Cell at Lowest Elevation Angle	(km)
11.	Reflectivity at Cell Top	(dBZ)
12.	Height of Cell Top	(km)
13.	Track Identifier	
14.	Smoothed Track Velocity - East	(m/s)
15.	Smoothed Track Velocity - North	(m/s)
16.	Integrated Tangential Shear	(m/s/km)

4. PROGRAM STATUS AND RECOMMENDATIONS

The processing programs are operational on the Interdata 7-32 computer. Experience must now be gained in using the system for the observation of weather. A number of parameters (Table 1) were set in the program on the basis of our experience with the data obtained from our contracts with the FAA and BuRec. It is anticipated that a different radar system operating in a different environment may need a different set of parameters. These parameters, such as the tracking measure weights and the constants in the velocity smoothing filter, are readily changed in the program. Experience with a larger data set is required to obtain the best estimate values for the parameters.

It is recommended that the cell detection and tracking programs be used on available data to develop the required operational experience to adequately use the new displays. The new output attributes are in a form that may be readily adapted to objective warning and forecasting systems.

5. REFERENCES

Boak, T.I.III, A.J. Jagodnik, Jr., R.B. Marshall, D. Riceman and M.J. Young (1977): "R & D Equipment Information Report, Tracking and Significance Estimator", Final Report, Contract No. F19628-77-C-0148; Raytheon Company; AFGL-TR-77-0259, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts.

Crane, R.K. (1976): "Radar Detection of Thunderstorm Hazards for Air Traffic Control, Vol. I Storm Detection", Project Report ATC-67, Vol. I, MIT Lincoln Laboratory, Lexington, Massachusetts.

Crane, R.K. (1977): "Parameterization of Weather Radar Data for Use in the Prediction of Storm Motion and Development", Final Report Contract No. F19628-76-C-0264, Environmental Research & Technology, Inc.; AFGL-TR-77-0216, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts.

Crane, R.K. (1978): "Development of Techniques for Short-Range Precipitation Forecasts", Final Report Contract No. F19628-77-C-0058, Environmental Research & Technology, Inc.; AFGL-TR-78-0005, Air Force Geophysics Laboratory, Hanscom AFB, Massachusetts.

APPENDIX A

Operating Instructions*
CSS Files*
TET Files*
Task Establishment Maps*
Definition of Variables
List of Arrays
List of Common Blocks

*Listings for Real-Time and Post-Mission Programs

REAL-TIME PROGRAM OPERATING INSTRUCTIONS

TO EXECUTE REAL-TIME CELL DETECTION AND TRACKING PROGRAM (ERT)

1. Compile each subroutine file:

   DCOMPILE ERT1:File Name*

2. Create an object file for each of the compiled subroutine files:

   DCALOBJ ERT1:File Name*

3. Delete old task file:

   DE ERT1:CRANE.TSK

4. Create a new task file:

   ESTAB ERT1:GARY

5. Dispose I/O devices by editing the CSS file:

   EDITR CREAL.CSS

6. Execute the program:

   CREAL

*List of subroutine file names:

TSEMAIN	ATRAK
TSEDATA	BTRAK
REALTM	COMPAR
INPARM	RESOLV
CONTOR	COMBIN
PEAKD	STRAK
TRACK	

```
*
07/19/79  16:56:49
***LISTING FOR CREAL.CSS
$N
$JOB
L .BG,ERT1:TSEPARAM
T .BG
AS4,ERT1:TSEPARAM.DAT
ST ,@1,@2,@3,@4,@5,@6,@9
L .LIB,ERT1:TSERTL.RTL
SE PA 2/0,1/145.0
L CRANE,ERT1:CRANE
T CRANE
AS3,CON:
AS5,CON:
AS8,ERT1:PLT
AS9,ERT1:DAT1024
AS10,PPRI:
AS7,PACK:TSEPPI.PAR
AS6,PR:
$IFNNULL @7
   AS5,@7
$ENDC
$IFNULL @7
   AS5,ERT1:TSEDEF.PAR
$ENDC
AS4,ERT1:TSEPARAM.DAT
ST
$EXIT
*
```

REAL TIME TET FILE

```
*
07/19/79  16:55:46
***LISTING FOR ERT1:GARY.TET
$N
LOG
JOB ERT
REMOTE
ES TASK
MXSPACE 2800
OPTIONS F
GET 400
PRIORITY 10,10
TCOM TASK/2/RW
TCOM ZSTORE/3/RW
TCOM DONE/4/RW
TCOM EXTRA/5/RW
IN ERT1:TSEMAIN
IN ERT1:TSEDATA
IN ERT1:REALTM
IN ERT1:INPARM
IN ERT1:CONTOR
IN ERT1:PEAKD
IN ERT1:TRACK
IN ERT1:ATRAK
IN ERT1:BTRAK
IN ERT1:COMPAR
IN ERT1:RESOLV
IN ERT1:STRAK
IN ERT1:COMBIN
RESOLVE ERT1:TSERTL
EDIT FVRTL
BU TASK,ERT1:CRANE
MAP
END
*
```

TASK-ESTABLISHMENT LOAD MAP   REAL TIME TASK ESTABLISHMENT MAP

```
DATE:07/19/79    TIME:16:52:01
JOB:ERT
**** CTOP=0242FE    UTOP=023E40    MIN PARTITION= 144 75K    ****


PROGRAM SEGMENTS:

  SEG  TYPE      NAME          SIZE
   0   IMPU                    143.75K
   3   TCOM   ZSTORE   TCM       4.25K
   5   TCOM   EXTRA    TCM       1.25K
  15   RTL    TSERTL   RTL       7.75K


PROGRAM LABELS:

 000100 TSEMAIN    000620 TSEDATA    003838 ERTRMAP    010A80 ICLOCK
 010B70 COS        010CB8 SIN        010DF8 .I         010EE8 .A
 011000 .R         011110 SQRT       011210 ALOG10     0112D8 ALOG
 011430 EXP        011588 FLOAT      0115A8 IFIX       0115F8 MOD2
 011640 ABS        011678 IABS2      011698 .H         011708 .S
 011798 .O1        0117A0 .O         011D48 .MES


TASK ENTRY-POINTS:

 000664 PPRDSC     00387C RMAP       004204 CONTOR     0049A4 PEAKD
 009BEC TRACK      00A0A4 ATRAK      00A87C BTRAK      00AAF4 COMPAR
 00BFAC RESOLV     00EFCC STRAK      010764 COMBIN     010A82 ICLOCK
 010B72 COS        010B8A .COS       010CBA SIN        010CD2 .SIN
 010DFA .I         010EEA .A         011002 .R         011112 SQRT
 01112A .SQRT      011212 ALOG10     01122A .ALOG10    0112DA ALOG
 0112F2 .ALOG      01131A .LOGRT2    011432 EXP        01144A .EXP
 01158A FLOAT      0115AA IFIX       0115FA MOD2       011642 ABS
 01167A IABS2      01169A .H         01170A .S         0117A2 .O
 0117B2 .O1        01192E .OI        011D4A .MES


LOCAL COMMON BLOCKS:

 011D68 CALB       011D78 MUSIG      011D80 READZ      011D88 SECTOR
 011D90 REFL       0129B0 CAL        0129C0 RUNSUM     0139D0 TLIS
 0139E8 CALR       016208 SWITCH     020098 AZ2        0200B0 AZM
 0200C8 PNTRS      0200D8 INTL       0200E8 ZLOOK      020258 ECONST
 020260 MAPPAR     020278 CNT        020288 DATA1      020610 DATA2
 021198 DATA3      0214A0 NVLIS      0214B0 FILTER     0214C0 KTA
 0214C8 CDRAYS     0222D0 CONST      022300 VPARM      022308 DVAL
 022310 CNTRS      022320 FLGS       022330 TMAX       022338 PWORK
 0223E8 FIXED      0223F8 PRSTOR     023AE0 THRESH     023AE8 CONPK
 023AF0 RSLV       023C78 COMB


LIBRARY ENTRIES:

 0F0002 .U         0F0052 .V         0F006A .P         0F00FA .Q
 0F0182 @R         0F02C8 @H         0F17E2 CONMSG     0F186A FLOAT2


TASK COMMON BLOCKS:

 030000 ZSTORE     050000 EXTRA
END
```

POST-MISSION PROGRAM OPERATING INSTRUCTIONS

TO EXECUTE POST-MISSION CELL DETECTION AND TRACKING PROGRAM (CRANE)

1. Compile each subroutine file:

   DCOMPILE ERT1:File Name*

2. Create an object file for each of the compiled subroutine files:

   DCALOBJ ERT1:File Name*

3. Delete old task file:

   DE ERT1:CRANE.TSK

4. Create a new task file:

   ESTAB ERT1:CRANE

5. Dispose I/O devices by editing the CSS file:

   EDITR CRANE.CSS

6. Execute the program:

   CRANE

*List of subroutine file names:

CRANE	RESOLV
INPARM	COMBIN
CONTOR	STRAK
PEAKD	
TRACK	
ATRAK	
BTRAK	
COMPAR	

POST-MISSION CSS FILE

**Listing of the CRANE.CSS File

```
$JOB
SE PA 1/140,2/0,3/0
L LIB,FRT1:TSFRT1.RTL
L CRANE,FRT1:CRANE
T CRANE
AS2,NULL:
AS3,PR:
AS4,NULL:
AS6,FRT1:User Defined Output File*
AS7,FRT1:User Defined Calibration File**
AS9,FRT1:User Defined Input Data File***
ST
$TERMJOB
$EXIT
```

*File to which program output is to be sent

**Disk file containing values for the parameters: IOUT, DBB, MAXV, MAXS, SLOPE, OLDATA

***Disk file generated by program "FRT"

```
TET/32 R02-03                    POST-MISSION TET FILE
JOB ERT
REMOTE
ES TASK
MXSPACE 2800
OPTIONS F
GET 400
PRIORITY 10,10
TCOM TASK/2/RW
TCOM ZSTORE/3/RW
TCOM DONE/4/RW
TCOM EXTRA/5/RW
IN ERT1:CRANE
IN ERT1:INPARM
IN ERT1:CONTOR
IN ERT1:PEAKD
IN ERT1:TRACK
IN ERT1:ATRAK
IN ERT1:BTRAK
IN ERT1:COMPAR
IN ERT1:RESOLV
IN ERT1:STRAK
IN ERT1:COMBIN
RESOLVE ERT1:TSERTL
EDIT FVRTL
BU TASK,ERT1:CRANE
MAP
```

POST-MISSION TASK ESTABLISHMENT MAP

```
OS32MT TASK-ESTABLISHMENT LOAD MAP
DATE 07/19/79     TIME 13:10:27
JOB:ERT
**** CTOP=01E5FE    UTOP=01E1F8    MIN PARTITION= 121 50K  ****


PROGRAM SEGMENTS:

   SEG  TYPE     NAME         SIZE
     0  IMPU                  120.50K
     3  TCOM  ZSTORE  TCM       4.25K
     5  TCOM  EXTRA   TCM       2.50K
    15  RTL   TSERTL  RTL       7.75K


PROGRAM LABELS:

  000100 ERTRMAP    00E850 COS        00E998 SIN        00EAD8 .I
  00EBC8 .A         00ECE0 .R         00EDF0 SQRT       00EEF0 ALOG10
  00EFB8 ALOG       00F110 EXP        00F268 FLOAT      00F288 IFIX
  00F2D8 ABS        00F310 IABS2      00F330 .S         00F3C0 .O
  00F968 .MES


TASK ENTRY-POINTS:

  001FD4 CONTOR     002774 PEAKD      0079BC TRACK      007E74 ATRAK
  00864C BTRAK      0088C4 COMPAR     009D7C RESOLV     00CD9C STRAK
  00E534 COMBIN     00E852 COS        00E86A .COS       00E99A SIN
  00E9B2 .SIN       00EADA .I         00EBCA .A         00ECE2 .R
  00EDF2 SQRT       00EE0A .SQRT      00EEF2 ALOG10     00EF0A .ALOG10
  00EFBA ALOG       00EFD2 .ALOG      00EFFA .LOGRT2    00F112 EXP
  00F12A .EXP       00F26A FLOAT      00F28A IFIX       00F2DA ABS
  00F312 IABS2      00F332 .S         00F3C2 .O         00F3D2 .OI
  00F54E .OI        00F96A .MES


LOCAL COMMON BLOCKS:

  00F988 TLIS       00F9A0 CONST      00F9D0 SWITCH     019860 AZ2
  019878 AZM        019890 REFL       01A4B0 PNTRS      01A4C0 INTL
  01A4D0 ZLOOK      01A640 ECONST     01A648 MAPPAR     01A660 CNT
  01A670 DATA1      01A9F8 DATA2      01B580 DATA3      01B888 NVLIS
  01B898 FILTER     01B8A8 KTA        01B8B0 CDRAYS     01C6B8 VPARM
  01C6C0 DVAL       01C6C8 CNTRS      01C6D8 FLGS       01C6E8 TMAX
  01C6F0 PWORK      01C7A0 FIXED      01C7B0 PRSTOR     01DE98 THRESH
  01DEA0 CONPK      01DEA8 RSLV       01E030 COMB


LIBRARY ENTRIES:

  0F0002 .U         0F0052 .V         0F006A .P         0F00FA .Q
  0F0182 @R         0F02C8 @H         0F186A FLOAT2


TASK COMMON BLOCKS:

  030000 ZSTORE     050000 EXTRA
END
```

VARIABLE LIST

CRANE & REALTM

+TL1 - lower threshold (dBZ)
+TL2 - higher threshold (dBZ)
+RQUANT - threshold quantization factor (dBZ)
+II &*ZX - intermediate values in dBZ conversion
+IEOFF - reflectivity offset to insure positive values
*DB - calibration constant
*BITVEL - constant used in calculation of velocity
*BITVAR - constant used in calculation of variance
*F4WORD - square of variance
*ELEVAT - raw elevation (encoder units)
*ELEVAA - elevation in degrees
+IELSN - elevation in degrees
+T - time (seconds from start of year)
+AZ - raw azimuth (deg)
*AZT - azimuth in degrees
*A - azimuth in radians
*AZCHK - azimuth in deg. + 359
+K - radial counter
*BGNA - start azimuth (deg)
*ENDA - stop azimuth (deg)
*SINA - sin(A)
*COSA - cos(A)
+NA - check on first azimuth of new scan
+NAC - offset multiplier for arrays in CONTOR & PEAKD
+IDAY -+DAY - Julian date data collected
+IHR - hour
+IMIN - minute } time scan begins
+ISEC - second
*DELTAZ - azimuth increment (radians)
*PHI - elevation angle (radians)
*COSPHI - cos(PHI)
*SINPHI - sin(PHI)
*EARTH - 6.4857E-5 (km^{-1})
*COSPH2 - $COSPHI^2$*EARTH
*AZLAST - azimuth of previous radial (deg)

+ - INTEGER*2

* - REAL

CONTOR

+IFLAG - intermediate print flag
*TLS - lower threshold (TL1 in CRANE)
*TATRMN - test on area
+NEMC - array addressing offset
+NCEL - cell counter
+IEMAX, +JMAX - array limits
+NEM1 - address variable
+NCL - maximum number of positions
+TL1 - lower threshold
+IEM - event counter on first threshold
+IEM2 - event counter on second threshold
+IPB - peak start location
+IP - peak counter
+IEVENT - event number

+ - INTEGER*2

* - REAL

PEAKD

+NBADR,+NCADR,+NBKA,+NCKA - address variables
+NAX,+NA - radial counter
+LM - number of variables in UP (Array)
+LMDP,+NAN,+NAN1,+LMM,+IDX,+NCLM,+LDBM,+LDX,+NPDP,+ID2 - address variables
*FQUANT - threshold quantization factor
+KOFST,+LIMT,+NIDP,+MXTR,+KMAX - Array limits
+IE - event number
+IEM - number of events on radial
+IEA,+KIE,+KIEM - address variables on IEM
+ICEST - event start position
+ICESP - event end position
+JEB,+KA,+KB - address variable on event and peak
+IPL - peak start location
+IP - peak stop location
+NTHRES - threshold counter for peak
+LDB - dB below peak value used to define peak
+IR1 - range to peak
+IU - dBZ value at IR1
+IT - dBZ above threshold at IR1 for LDB thresholds
+JMXDM - limit on IT
+KA - address variable on peak thresholds
+IPT - number of thresholds associated with peak
+JR - limit on IPT
+IBGN - first position in event
+IND - last position in event
+IU - reflectivity values within event
+KA - address variable on peak threshold values within event
+IMXJMX - limit on number of contours per radial
+IREG - start or stop range of contour
+IPE - contour counter
+IADDR & +IEQL - address variables on contour threshold and number
+KC,+KA,+KZ - address variables on contour thresholds
+TCVL,+TCVM,+TCVLB - threshold values on a peak
+NPC - number of contours by threshold on this radial
+NPL - contour counter by threshold
+IHBM - start range of contour
+IHB - IHBM + 1
+IHD - stop range of contour
+K,+KY,+KZ - address variables on next threshold
+LPE - number of contours, this threhold, on radial
+LPL - contour counter, this threshold
+NPCEL - ID number for possible cell, this threshold
+TATC - ID TATR(NPCEL) points to
+JE1 - first event previous radial
+JE2 - last event previous radial
+JEM - address variable on events, previous radial
+IPB - number of peak thresholds
+KB,+KBB,+KBA,+KBC - address variables on contour thresholds
+TBVL - threshold value on a peak
+NP2 - numb'r of contours on a threshold, previous radial
+NP1 - cont)ur counter, by threshold, previous radial

+ - INTEGER*2   * - REAL

PEAKD (continued)

+LPCEL - ID number for possible cell, previous radial
+IEQL - ID TATR(LPCEL) points to
+MPC - equals NPCEL if associated
+JEQL - next higher threshold
+JN1,+JN2,+JN3,+JN4,+JN5 - address variable on peak
+IST - start range of contour
+ISP - stop range of contour
*R - area per azimuth degree at peak range
+IU - reflectivity at that range
*RU - reflectivity weighted area
*SAZ - sin(azimuth)
*CAZ - cos(azimuth)
+KNN,+KN - address variable on NPCEL
+IMDX - address variable on second threshold of peak
+IND,+INDX,+LNX - address variable on LPCEL
+IN,+IEQL - address variable on LPCEL
+JEQL - area address of NPCEL pointed to by area of LPCEL
+IPTT - number of peak thresholds
+KTI,+KTA,+KTB - address variables on contour threshold
+NPCT - number of contours, this threshold
+IEQL - threshold value
+INDXT - address variable on NPCEL
+NIMN - number of possible cells tested so far
+NIDP - limit on number of possible cells
+IE - event number
+IPT - number of peak thresholds in event
+NPC - number of thresholds this peak
+NCVM - threshold value on peak
+NPL - contour counter

+ - INTEGER*2

* - REAL

PEAKD (continued)

+I - each position out the radial
+IA - address variable on I
+IEQL - ID value at each position
+J - event number on previous radial
+JA - address variable on J
+IPB - number of peak thresholds in event
+KA,+KAP,+KAM - address variables on event and threshold
+NP - number of contours, this threshold, on previous radial
+ITERM - code for eliminating possible cell
+MG - address in UP (array) for measure of cell significance
+LMT - limit on number of cells to be carried in order of significance
+JK1 - address variable on previous radial
+JKL - address variable on current radial

+ - INTEGER*2

TRACK

*VKM - cos (elevation) x conversion m to km
*SAVKM - unit area convert from m^2 to km^2
+MA - address in ECL (array) of measure of cell significance
+M - cell counter
+M1,+M2,+M3,+M4,+M5,+M6,+M7 - address of each attribute in ECL (array) by cell
+KOFST,+NAN2 - array addressing offset
*VKME - convert reflectivity weighted line of sight distance to horizontal dist.
*FNSN,+NSCAN - scan counters
+KTL - time
+JDAY,+IDAY - Julian date
+JHR,+IHR
+JMIN,+IMIN start time of first scan in sequence
+JSEC,+ISEC
+NC - cell number (current scan)
+NCG,+NCBG,+NCB - NC to pass through common
+NVMIN,+NVMX,+NCMX - number of significant cells detected this scan
+IELSN - current elevation angle
+IESNL - elevation angle last scan

ATRAK

+NCEC - address offset on NC
+NVVC - address offset on NV
+NC - cell counter for ECL (array)
+NV - cell counter for VCL (array)
+NCA - address variable on NC
+NVA - address variable on NV
+IZ - cell reflectivity
*X - cell position east [(-)west]
*Y - cell position north [(-)south]
*H - cell height
+IZL - offsetted reflectivity
*Z - reflectivity
*HL - height of last cell
+IZP - peak reflectivity

BTRAK

+NCEC - address offset on NC
+NVVR - address offset on NV
+NC - cell counter for ECL (array)
+NV - cell counter for VR (array)
+NVA - address variable on NV
+NCA - address variable on NC

+ - INTEGER*2

* - REAL

COMPAR

+NCMX - number of cells detected this scan
+IM,+JM - array limits
+NC - cell numbers detected this scan
+NCEC - address offset on NC
+(NC1 to NC6) - address variables on NC
+NV - cell numbers tracked from previous scans
+NVVC - address offset on NV for VCL (array)
+NVVR - address offset on NV for VR (array)
+NLR - address variable on NV
*ATEST - estimate of cell NV's movement from last scan to this scan
*DELT - time since last scan
*DELX, DELY - distance on X & Y coordinates between cell NV and NC
*DELW - a measure of the association between NV and NC using reflectivity,
location, area, height
+JO - overflow of D and ID (arrays)
+NSCAN - scan number
+IO - overflow of IUV (arrays)
*DX - DELW of a previous association with this NC cell
+NVT - NV previously associated with this NC cell
+NCT - NC previously associated with this NV cell
*DX - DELW of a previous association with this NV cell
+NVMX - number of active cell tracks
+NCR - NC cell to test NV cells against
+NVB - NV } associated
+NCB - NC
*HCT - update height

+ - INTEGER*2

* - REAL

RESOLV

*NVT - NV associated with NCT
+I - number of NV associations on this NCT
+JX - number of NC cells associated with 1 NV cells
*NCT - NC associated with NVT
+I - number of NC associations on this NVT
+JX - number of NV cells associated with 1 NC cells
#NVT - NV cell that NC cell is associated with (highest DELW - down)
+JV - number of cells associated to this NV cell
#KV - maximum number of associations to one NV cell
#IVS - NV cell association on
+LC - counts NC cells checked
#NCT - same as NVT but NC on NV
+JC - same as JV but on an NC cell
#KC - same as KV but on an NC cell
#ICS - NC cell association on
+LV - counts NV cells checked
+NV - NV cell that NC cell is associated with
+KA,+KA2,+KA1 - address variables on KC
+NC - NC cell that NV cell is associated with
+IMSM - counts associations on NV other than NC
*DELW - measure of compare
*DWT - minimum measure of compare
+NV - cell with DWT
+KNC - counts cells with too many associations
*DELT - test on cell velocity
+NVT1 - next NV cell associated with NC (in order of DELW)
*DWT1 - minimum measure on NVT1
+NV1 - cell with DWT1
*DELW1 - test on NV1
*DELW2 - test on NV
*HTE - update height

+ - INTEGER*2

- INTEGER*4

* - REAL

STRAK

+NVA - address offset on NV
+(NV1 to NV41) - address variables on NV for VCL (array)
+NVR - address offset on NV
+(NR1 to NR6) - address variables on NV for VR (array)
+NV - cell counter
*VXT - velocity east [(-)west]
*VYT - velocity north [(-)south]
*DELTM - time since last scan
+IZVAL - reflectivity
+IDTC - percent of scans cell was detected
*VXC - sum of eastward velocity components
*VYC - sum of northward velocity components
+NFN - number of scans processed
+NSN - number of velocity values summed in VXC and VYC
*VX - average eastward velocity of all cells
*VY - average northward velocity of all cells
+NVSCN - number of volume scans
+KTL - time of last scan

+ - INTEGER*2

* - REAL

LIST OF ARRAYS - THEIR SIZE AND CONTENT

In CRANE

ANC(1028) - header information - INTEGER*4
ZEE(1024) - raw data - INTEGER*4
IREF(1024) - reflectivity - INTEGER*2
IVEL(1024) - radial velocity - INTEGER*2
IVAR(1024) - variance - INTEGER*2

RE(1025) - decoded reflectivity - INTEGER*2
IVELL(256) - decoded velocity - INTEGER*2
IDVEL(256) - tangential shear - INTEGER*2
ZARY(91) - convert refelctivity to dBZ - REAL

In CONTOR

ICL(44)* - start position of event - INTEGER*2
IC2(44)* - stop position of event - INTEGER*2

IDC(22) - number of peaks in each event - INTEGER*2
IPRNG(34) - location of peaks - INTEGER*2

IC21(22) - start position of event on second threshold - INTEGER*2
IC22(22) - stop position of event on second threshold - INTEGER*2

*array contains indicated parameter(s) on the current radial, offsetted from the same parameter(s) on the previous radial

In PEAKD

T(80) - all possible thresholds a peak may have - INTEGER*2
TC(1980)* - thresh of each peak - INTEGER*2
IPTC(44)* - number of thresholds in each event - INTEGER*2
IPCNT(1980)* - contour counter - INTEGER*2
IPC1(5400)* - start range of contour segment - INTEGER*2
IPC2(5400)* - end range of contour segment - INTEGER*2
IPC3(5400)* - number of peaks within the segment - INTEGER*2

TATR(1400) - temporary attribute array - stores peak attributes until a cell is detected or peak discarded - REAL
IACT(70) - overflow (too many peaks) - INTEGER*2

UP(6) - cell attributes - REAL

1 - area
2 - reflectivity
3 - location in km east of radar
4 - location in km north of radar
5 - tangential shear
6 - a measure of relating cell significance = const * area + tan shear

*array contains indicated parameter(s) on the current radial, offsetted from the same parameter(s) on the previous radial

PEAKD & TRACK: ECL (7x16x2) (I,N,NAN1)

I	variable		
1	area surf	real	
2	dBZ	real	
3	east	real	
4	north	real	
5	range	real	
6	height	real	defined in TRACK
7	tang shear	real	

ATRAK: VCL (23x32) (I,N)

I	variable			
1	east (X)	real		
2	north (Y)	real		
3	dBZ	int*4		
4	area surf	real	set in ATRAK	lowest elevation
5	time	int*4	def in CRANE	angle only
6	height	real		
7	range	real		
8	NT; track ID	int*4	def in ATRAK	
9	Σ #scans each det.	int*4		
10	ΣZ	real		
11	ΣZX	real		
12	ΣZY	real		updated each
13	ΣZ(H-HL)xArea	real		scan
14	H summit (HT last)	real		
15	dBZ summit	int*4		
16	dBZ peak	int*4		
17	H peak	real		
18	$\overline{X}$	real	def in STRAK	
19	$\overline{Y}$	real		updated each
20	time	int*4		vol scan
21	Vel x	real		
22	Vel y	real		
23	Σ tang shear	real	def in ATRAK	updated each scan

STRAK: TCL (21)   (I)

I	variable	
1	time	int*4
2	$\overline{X}$	real
3	$\overline{Y}$	real
4	$\overline{Z}$	real
5	(H-HL)xArea	real
6	(not used)*	
7	dBZ peak	int*4
8	H peak	real
9	(not used)	
10	dBZ	int*4
11	area surf	real
12	height	real
13	(not used)	
14	(not used)	
15	dBZ summit	int*4
16	H summit	real
17	ABS(NT) TRAKID	int*4
18	(not used)	
19	(not used)	
20	$\Delta\overline{X}/\Delta T$	real
21	$\Delta\overline{Y}/\Delta T$	real

*array locations not used in the current version of the program

In COMPAR

track cells from previous scan (NV) to cells in current scan (NC)

IUC1(16) - points to NV each NC is associated with - INTEGER*2
i.e. IVC1(NC) = NV

IC(16) - stores the "measure" of the above association - INTEGER*2

IUC2(16) - if one NC is associated with more than one NV then IVC2(NC) $\neq$ 0 - INTEGER*2

IUV1(32) - points to NC each NV is associated with - INTEGER*2

IV(32) - stores the "measure" of the above association - INTEGER*2

IUV2(32) - if one NV is associated with more than one NC then IUV2(NV) $\neq$ 0 - INTEGER*2

IC(32x10) - if IUC2(NC) $\neq$ 0 - INTEGER*2
IC(#,1) = number of conflicts
IC(#,2-10) = the NV's associated with

C(32x9) - stores the measures of each NC-NV association - REAL

ID(32x10) - same as the IC(array) for conflicts on NV - INTEGER*2
D(32x9) - stores the measures of each NV-NC association - REAL

In RESOLV

IV(32,7) - ordered IC(array) or ID(array) - INTEGER*2
V(384) - stores the measures of the tested associations - REAL

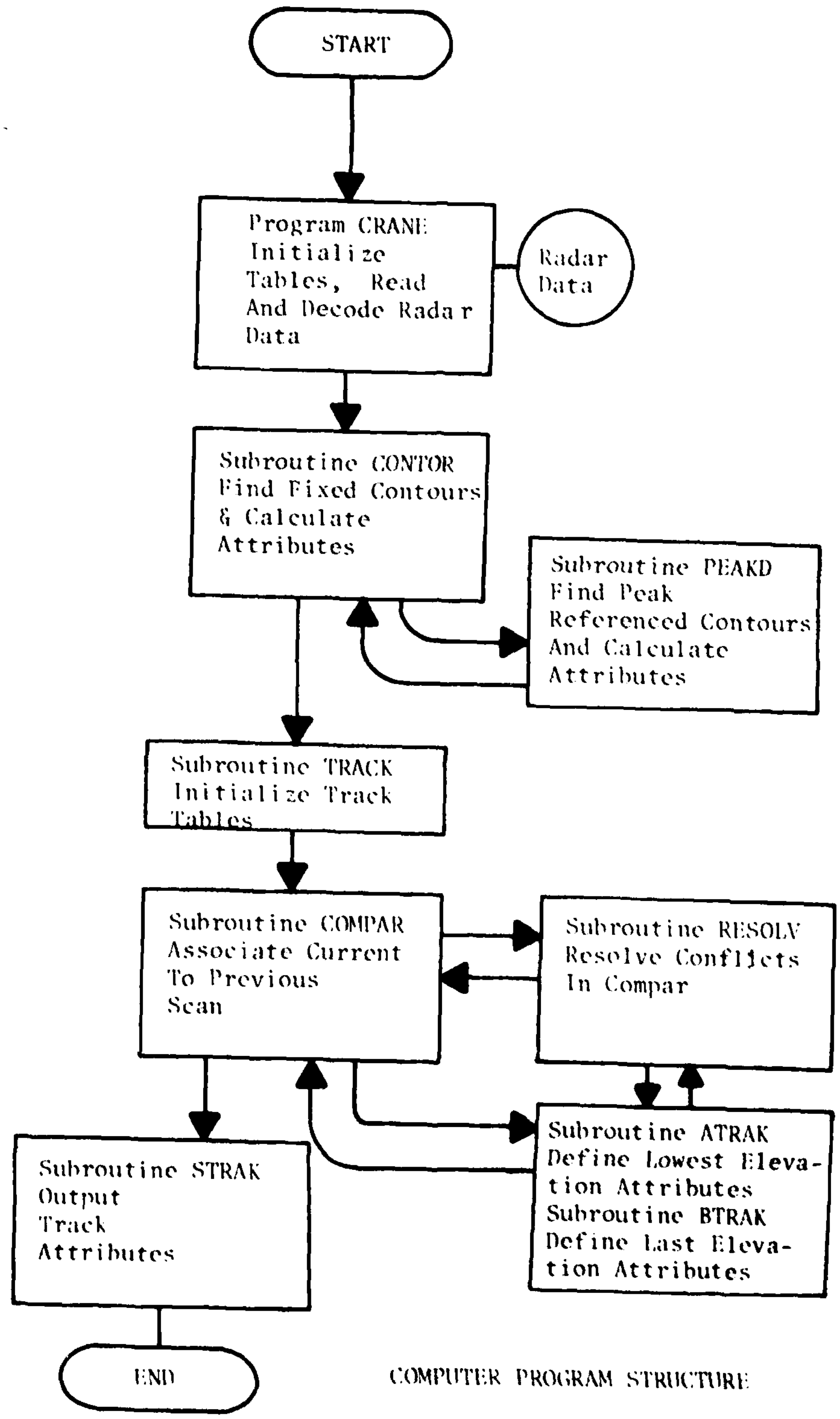
LIST OF COMMON BLOCKS AND THEIR ASSOCIATED ROUTINES

BLOCK NAME/SIZE*	CRANE	BLOCK DATA	CONTOR	PEAKD	TRACK	ATRAK	BTRAK	COMPAR	RESOLV	STRAK	COMBIN
AZM (4.5)	X	X	X	X							
AZ2 (6)	X	X	X	X	X						
CDRAYS (897)		X						X	X		
CLST (257)	X	X			X	X		X	X	X	
CNT (3)	X				X			X	X		
CNTRS (3)		X									
CONPIC (1)			X	X							
CONST (12)		X						X	X	X	
DATA1 (226)	X	X		X	X	X	X	X		X	
DATA2 (737)	X	X				X		X	X	X	
DATA3 (193)	X	X					X	X		X	
DVAL (1)		X				X					
ECONST (2)	X	X			X						
EXTRA (597.5)	X										
FILTER (3)	X	X	X	X							
FIXED (2.5)		X	X	X							
FLGS (3.5)		X				X		X	X	X	
INTL (3)	X	X			X	X		X	X	X	
KTA (2)					X	X		X	X		
KTB (1)					X		X	X	X		
MAPPAR (4.5)	X										
NVLIS (4)	X	X			X	X	X	X	X	X	
PNTRS (4)	X	X			X	X		X	X	X	
PRSTOR (1465)		X	X	X							
PWORK (43.5)		X	X	X							
REFL (775)	X	X	X	X							
RSLV (72.5)								X	X		
SWITCH (1047.5)	X		X	X							
THRESH (.5)		X	X	X							
TLIS (5)	X				X	X	X	X		X	
TMAX (.5)		X								X	
VPARM (2)		X						X		X	
ZLOOK (91.5)	X	X				X					
ZSTORE (1028)	X										

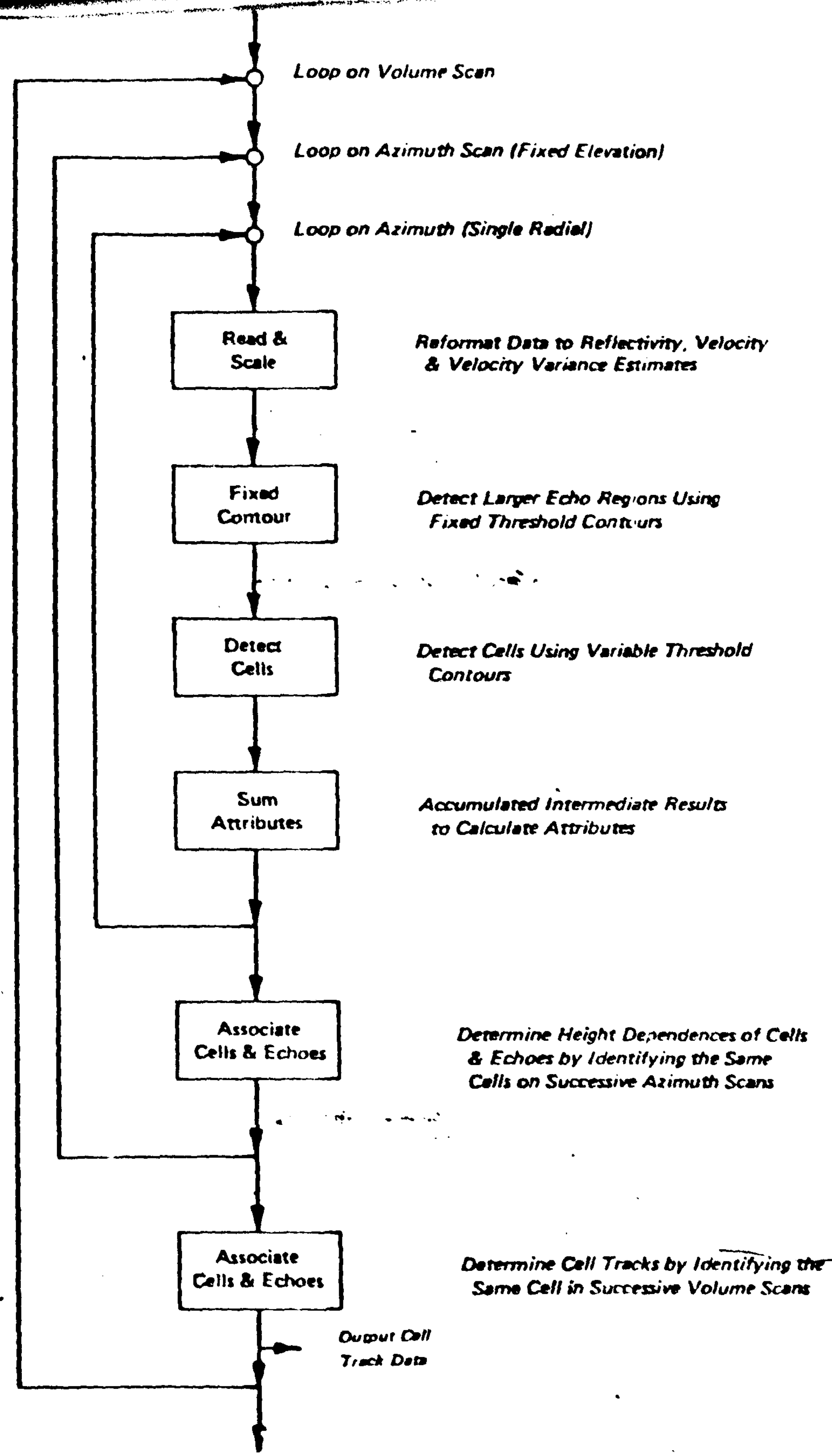
*size is number of 32 bit words

APPENDIX B

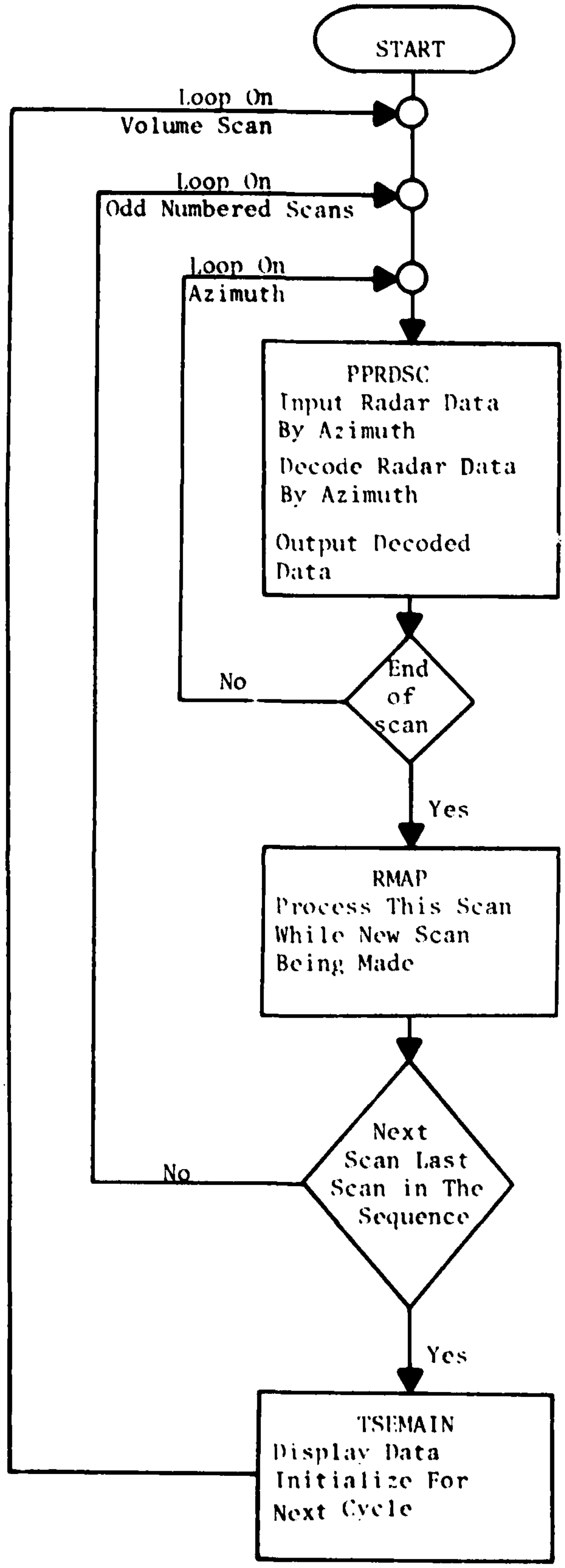
FLOW CHARTS



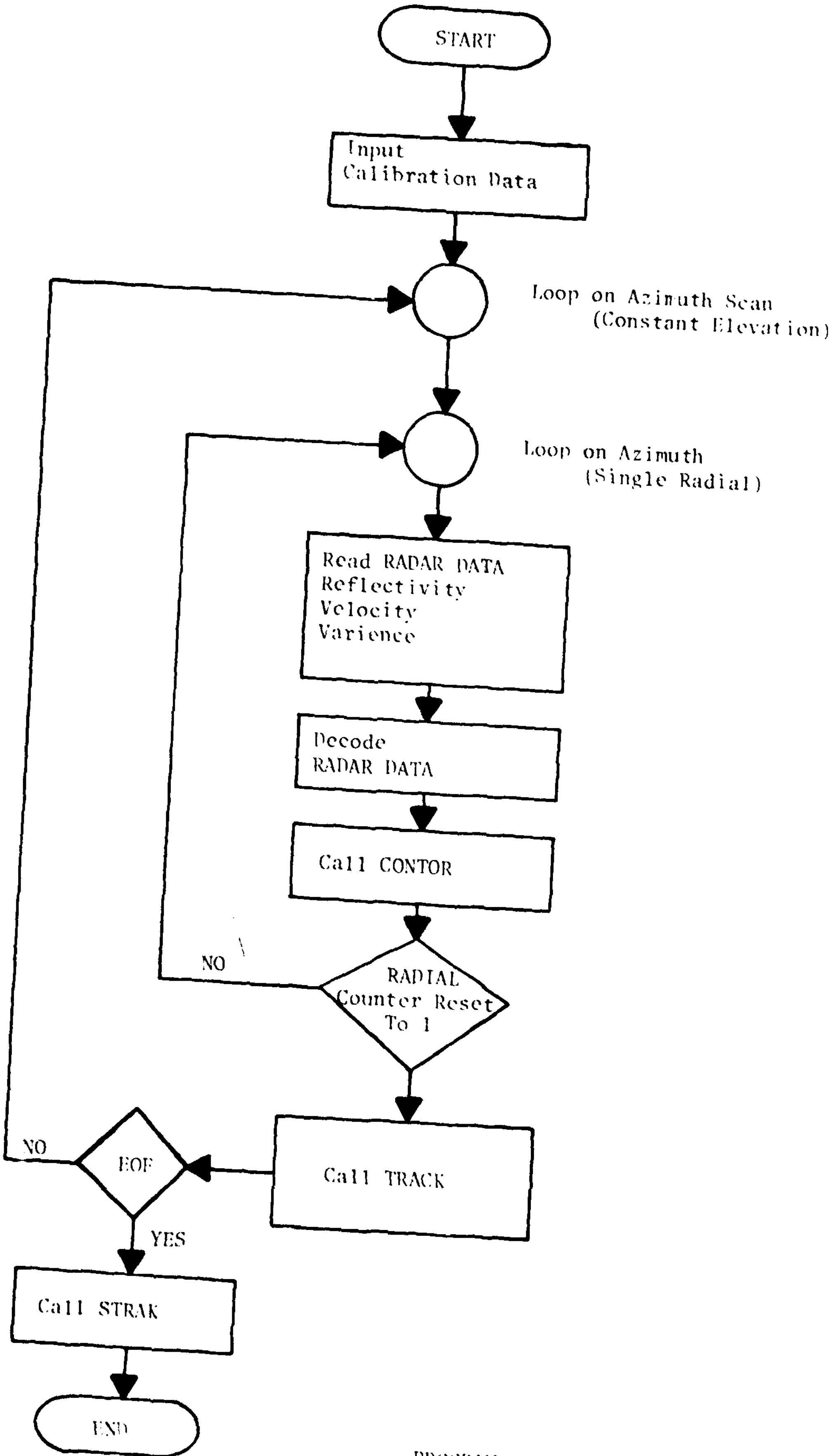
COMPUTER PROGRAM STRUCTURE



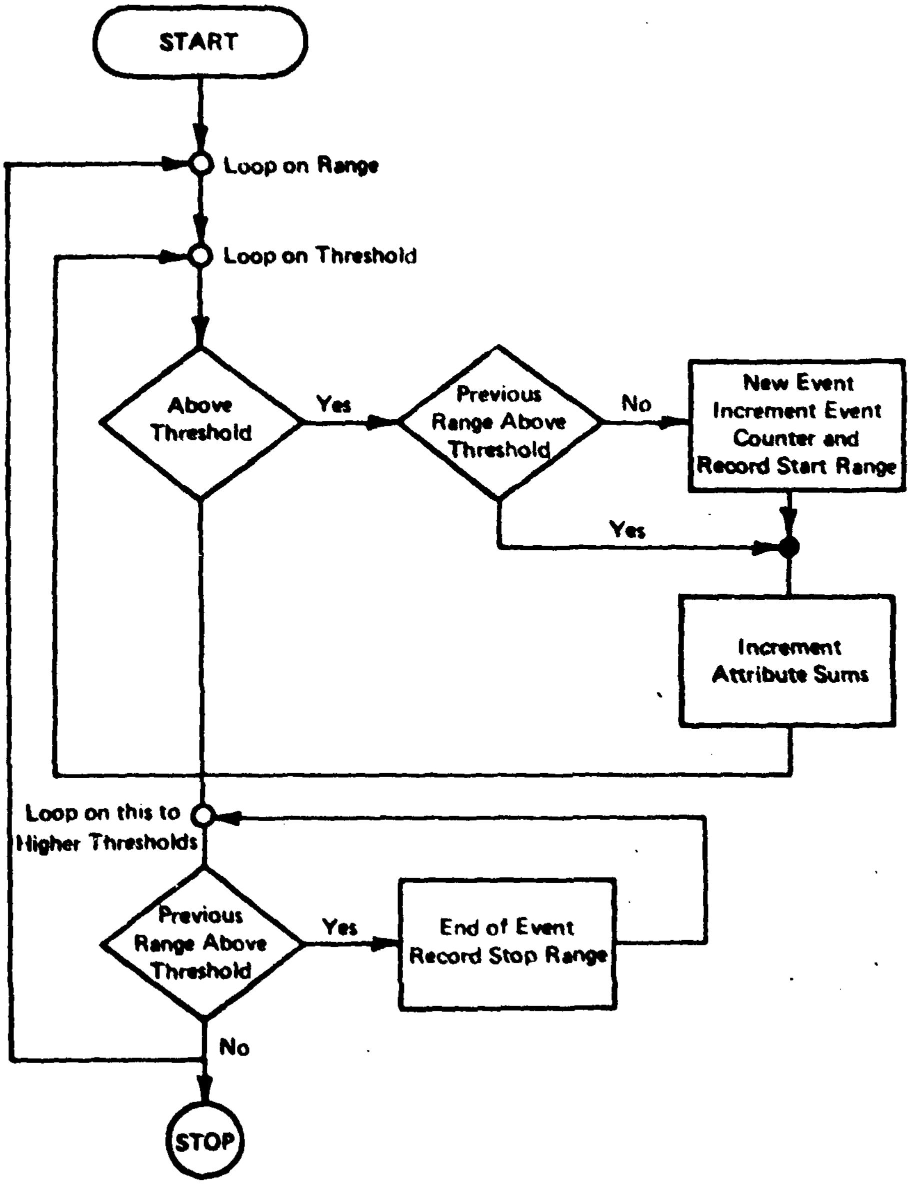
Overall Processing Scheme



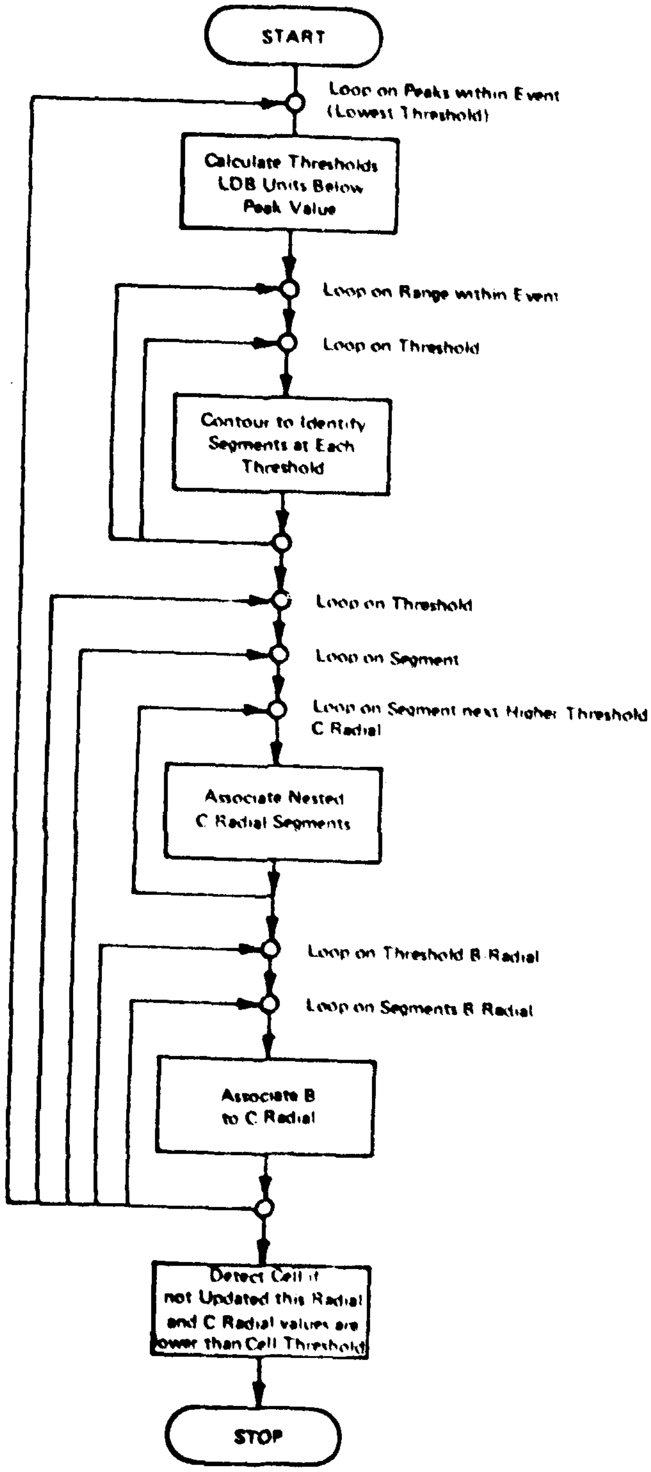
REAL TIME PROCESSING SCHEME



PROGRAM CRANE

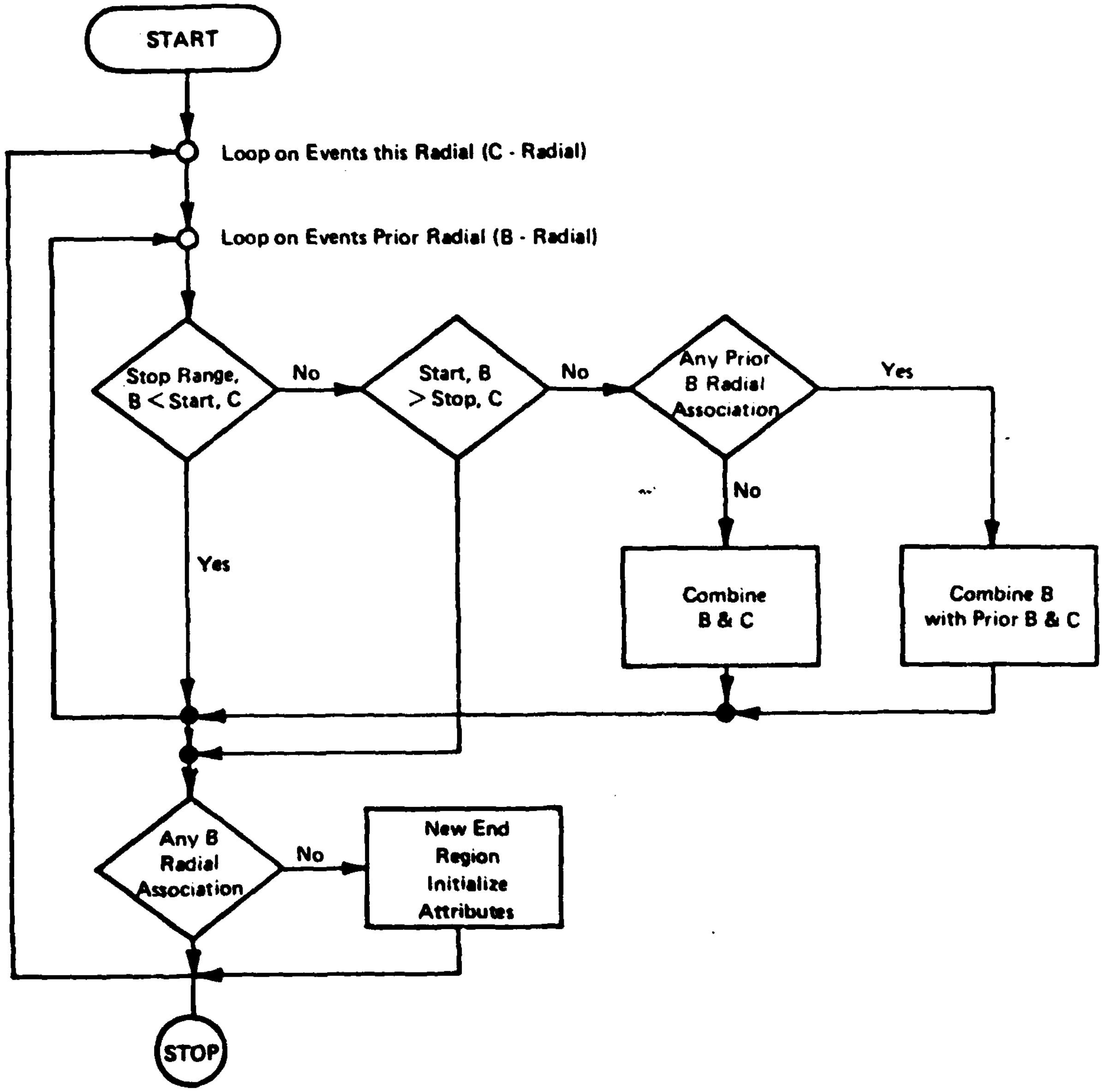


Event Identification
Subroutine CONTOR



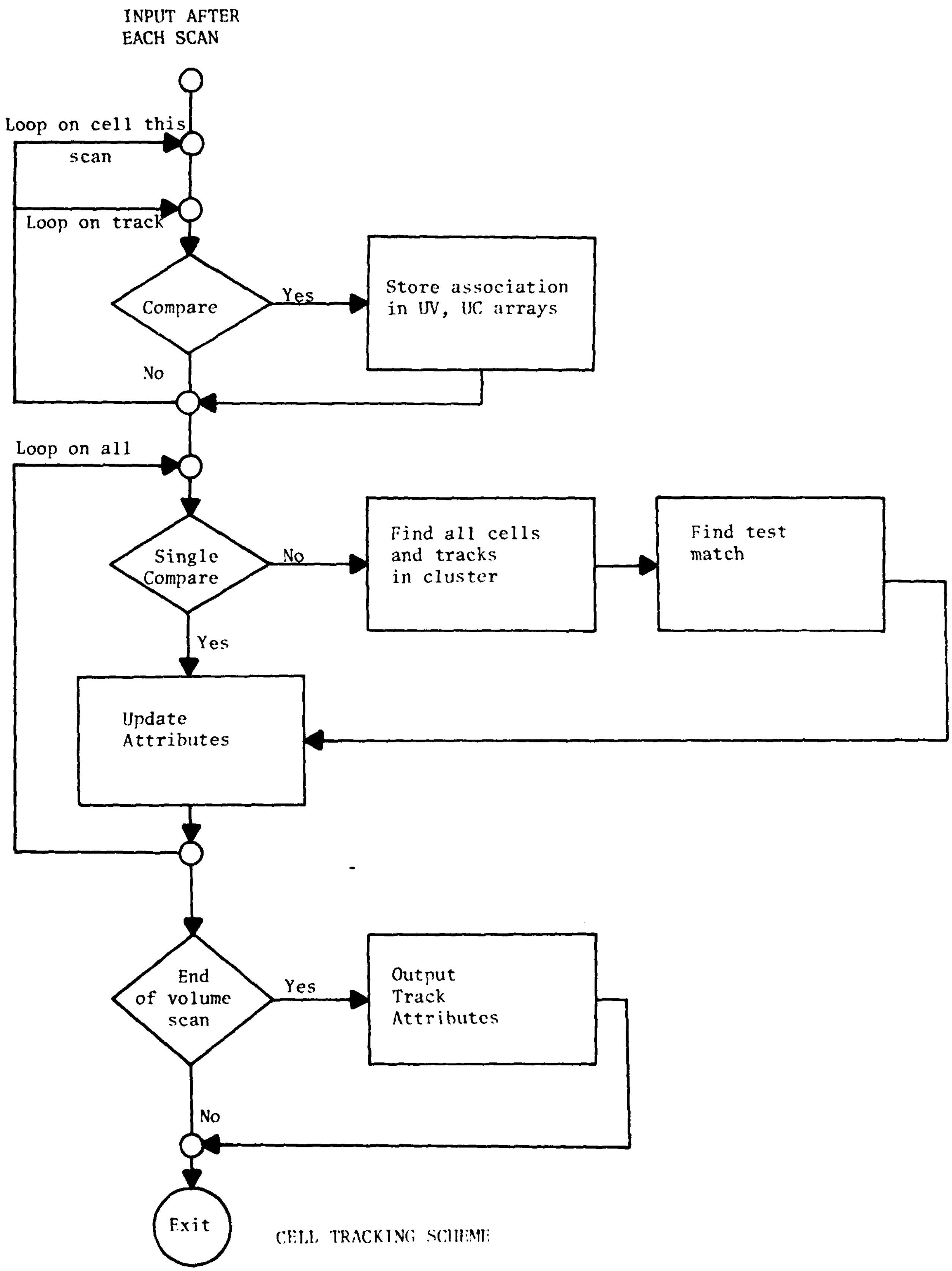
Peak Detection

Subroutine PEAKD



Event Association

Subroutine PEAKD



CELL TRACKING SCHEME

APPENDIX C

REAL-TIME VERSION

```
*
07/19/79  16:31:48
***LISTING FOR ERT1:TSEMAIN.FTN
$N
$ASSM
          NORX3
TSEMAIN  PROG
$FORT
$TITL FILE TSEMAIN - MAIN PROGRAM FOR ETSE
      IMPLICIT INTEGER*2 (A-Z)
      REAL DB,BITVEL
      INTEGER*2 RHO,TFSO,GRND,I,ZERO,T,TSY,STOP,ZTH,TFS,K,BETA
      INTEGER*2 INDCTR,ANG(7),DISPLA,MMU,SIGMA,ELEV(2)
      INTEGER*4 TIME,RSAVE(16),SECOND,NRCEAD
      INTEGER*4 NEXTIM,ITIME
      REAL PCTMIN,RRAREA
      COMMON /CALB/ DB,BITVEL,NRC
      COMMON /MUSIG/PCTMIN,MMU,SIGMA
      COMMON /READZ/NRCEAD,OLD
      COMMON /SECTOR/INDCTR
      COMMON /EXTRA/RHO,GRND,ZTH,BETA,K,RRAREA(12,24),RHO2,ZTH2,BETA2,
     1PCT2,MMU2,SIGMA2,MINUT2,ANG2(6),CHANGE
      EQUIVALENCE (DISPLA,ANG(1)),(DEMAND,ANG(6)),(ELEV(1),ANG(4))
      EQUIVALENCE (ANGLE,ANG(2)),(OLDATA,ANG(7))
      DATA DISPLA/4/,NLEVEL/3/
      REWIND 4
      INDCTR=0
10000 CONTINUE
$ASSM
          FREZE
          COPY  SVC1.
          STM   0,RSAVE
READ      SVC   1,READBLOK          READ IN DISPLAY, ANGLES
          LH    R0,READBLOK+SVC1.STA READ IN DEVICE STATUS
          BNZ   IOERR               BRANCH IF NOT ZERO
          LH    R0,ANGLE            LOAD IN BEGINNING ANGLE
          CHI   R0,360              LESS THAN 360?
          BNL   ALLD                NO, FULL SCAN
          LIS   R0,1
          STH   R0,INDCTR           INDCTR=1
ALLD      LM    0,RSAVE
$FORT
      READ(7,333) IOUT,DBB,MAXV,MAXS,SLOPE
  333 FORMAT(I3)
      BITVEL=MAXV
      DB=65.28
      READ(5,110)RHO,ZTH,BETA,IPCTMN,MMU,SIGMA,MINUTE
110   FORMAT(I3)
      CHANGE=0
      PCTMIN=IPCTMN*0.01
      SECOND=MINUTE*60
      OLD=OLDATA
50    CONTINUE
      CALL ICLOCK(2,TIME)
      NEXTIM=TIME+SECOND
      SVELE=ANG(5)
      ANG(5)=ANG(4)+5
      DO 51 N=1,NLEVEL
      CALL PPRDSC(ANG)
      CALL RMAP
      ANG(5)=SVELE
   51 CONTINUE
      CALL PPRDSC(ANG)
```

```
C       CALL TSEPLT(DISPLA)
        IF(CHANGE)55,55,52
        CHANGE=0
52      IF(RHO2.NE.(-1))RHO=RHO2
        IF(PCT2.NE.(-1))IPCTMN=PCT2
        IF(MMU2.NE.(-1))MMU=MMU2
        IF(SIGMA2.NE.(-1))SIGMA=SIGMA2
        IF(ZTH2.NE.(-1))ZTH=ZTH2
        IF(BETA2.NE.(-1))BETA=BETA2
        IF(MINUT2.NE.(-1))MINUTE=MINUT2
        DO 54 I=1,6
        IF(ANG2(I).NE.(-1))ANG(I)=ANG2(I)
54      CONTINUE
        PCTMIN=IPCTMN*0.01
        SECOND=60*MINUTE
55      IF(DEMAND-1)56,75,50
56      CALL ICLOCK(2,ITIME)
        IF(ITIME.GT.TIME)GO TO 58
        IF(NEXTIM.LE.86400)GO TO 50
        NEXTIM=NEXTIM-86400
58      IF(NEXTIM.LT.ITIME)GO TO 50
60      CONTINUE
$ASSM
R0        EQU    0
R1        EQU    1
R2        EQU    2
R3        EQU    3
R4        EQU    4
          STM    R0,RSAVE
          L      R1,NEXTIM
          ST     R1,SVCTIM          STORE TIME OF DAY
          SVC    2.TIMBLK           TIME OF DAY WAIT
          LM     R0,RSAVE           RETURN TO FORTRAN
$FORT
        GO TO 50
75      PAUSE 0
        GO TO 50
$ASSM
          ALIGN 4
IOERR     EXBR   R1,R0
          NHI    R1,X'00FF'
          CHI    R1,X'88'           END OF FILE?
          BE     ALLD               YES, FULL SCAN,DEFAULT DISPLAY
          CHI    R1,X'90'           END OF MEDIUM
          BE     ALLD               YES,FULL SCAN,DEFAULT DISPLAY
          SVC    2.ERRCODE
          SVC    2,ERRBLOK          PRINT OUT ERR MSG TO CONSOLE
          SVC    2,PAUSE            PAUSE
          B      READ               TRY AGAIN
          ALIGN 4
READBLOK  EQU    *
          DB     X'58'              READ BINARY AND WAIT
          DB     4                  LU 4
          DC     H'0'               STATUS
          DC     A(DISPLA)             BEGINNING OF BUFFER
          DC     A(OLDATA+1)         END OF BUFFER
          DSF    3
          ALIGN 4
ERRCODE   EQU    *
          DB     0                  4 CHRS OF ASCII HEX
          DB     6
          DC     H'0'
          DC     A(ERCD)            TO ERROR MSG
          ALIGN 4
ERRBLOK   EQU    *
```

```
        DB     7
        DC     H'30'                    30 CHRS
ERCD    DC     C'        '
        DC     C'I/O ERROR IN RTN TSEMAIN'
        ALIGN 4
PAUSE   EQU    *
        DB     0,1
        ALIGN 4
TIMBLK  EQU    *
        DB     0
        DB     10                       CODE 10
        DB     0,0
SVCTIM  DC     F'0'                     TIME OF DAY
$FORT
     END
*
```

```
*
07/19/79  17:07:53
***LISTING FOR ERT1:TSEDATA.FTN
$N
$ASSM
          SCRAT
          SQUEZ 3
TSEDATA   PROG
$FORT
$TITL FILE TSEDATA - DATA INPUT SUBROUTINE FOR ETSE
      SUBROUTINE PPRDSC(ANG)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*4 PPRI(1028),PPRI2(1028),RSAVE(16)
     +,RAW,DECOD(1028),DECOD2(1028)
      REAL DB,BITVEL
      INTEGER*2 IREF(1024),IVEL(1024),IVELL(1024),
     +IREF2(1024),IVEL2(1024)
      INTEGER*4 PPRANG
      INTEGER*2 ANG(6),RE(1025),HR(258),TL1,TL2,RQUANT
      EQUIVALENCE (PPRANG,PPRI(3)),(PP,PPRI(2))
      EQUIVALENCE (DECOD(5),IREF(1)),(DECOD(517),IVEL(1)),
     +(DECOD2(5),IREF2(1)),(DECOD2(517),IVEL2(1))
      COMMON /REFL/ RE,HR,NCL,NID,NIDP,INCL,IMX,
     +IMN,TL1,TL2,RQUANT,IDVEL(258)
      COMMON /CAL/DB,BITVEL,NRC
      COMMON /SECTOR/INDCTR
      COMMON /ZSTORE/PPRI
      COMMON /RUNSUM/PPRI2
      CALL CONMSG(6,'PPRDSC')
2     REWIND 9
      NAZ=0
      ELEV1=ANG(4)*11.37778
      ELEV2=ANG(5)*11.37778
      IF(INDCTR.EQ.2)INDCTR = 0
      IF (INDCTR .EQ. 0)GO TO 1
      CW = ANG(3) - ANG(2)
      IF (CW .GT. 180) CW = CW - 360
      IF (CW .LT. (-180)) CW = 360 + CW
      IF (CW .LT. 0) CW = 0
      IF (CW .GT.0) CW = 15
      BGNA = 11.37778 * ANG(2)
      ENDA = 11.37778 * ANG(3)
      BGNA=MOD(BGNA,4096)
      ENDA=MOD(ENDA,4096)
      IF(ENDA.GT.4086)ENDA=0
      STPFLG = 0
      NITG=4
      NRC=768
      IF (CW .GT. 0 .AND. ENDA .LT. BGNA) STPFLG = 1
      IF (CW .EQ. 0 .AND. ENDA .GT. BGNA) STPFLG = 1
      IF(BGNA.GT.10 .AND. BGNA.LT.4086)GO TO 1
      BGNA=0
      STPFLG=2
 1    CONTINUE
10000 CONTINUE
$ASSM
          FREZE
          CROSS
          COPY  SVC1.
R0        EQU   0
R1        EQU   1
R2        EQU   2
R3        EQU   3
```

```
R5       EQU   5
R6       EQU   6
R14      EQU   14
R15      EQU   15
R13      EQU   13
R6       EQU   6
         STM   0,RSAVE
         L     R4,WAITREAD+SVC1.SAD
         AIS   R4,15
         ST    R4,WAITREAD+SVC1.EAD
         BAL   R13,WREAD
         LIS   R1,4
         L     R0,PPRI(R1)
         NHI   R0,3
         AIS   R0,1
         SLLS  R0,8
         AIS   R0,4
         SLLS  R0,2
         SIS   R0,1
         L     R4,PPRIBLK+SVC1.SAD GET BEGINNING ADDRESS
         AR    R4,R0              COMPUTE END ADDRESS
         ST    R4,PPRIBLK+SVC1.EAD STORE IN EAD
         L     R4,WAITREAD+SVC1.SAD
         AR    R4,R0
         ST    R4,WAITREAD+SVC1.EAD SAME NUMBER
         L     R4,OUTBLK+SVC1.SAD
         AHI   R4,1028*4-1
         ST    R4,OUTBLK+SVC1.EAD
         L     R4,PPRIBLK2+SVC1.SAD GET NEXT BEGINNING ADDR
         AR    R4,R0              COMPUTE END ADDRESS
         ST    R4,PPRIBLK2+SVC1.EAD STORE IN SVC BLOCK
         L     R4,OUTBLK2+SVC1.SAD
         AHI   R4,1028*4-1
         ST    R4,OUTBLK2+SVC1.EAD
         LDAI  R15,PPRI
         LDAI  R14,PPRI2
         LCS   R6,1                 SET R6 TO -1 FOR COUNTER
DETECT   BAL   R13,WREAD            READ IN AN AZIMUTH
         LH    R1,8(R15)            LOAD IN AZIMUTH DATA
         NHI   R1,X'FFF'            AND OUT UNWANTED BITS
         BAL   R13,WREAD            GET ANOTHER AZIMUTH
         LH    R2,8(R15)            GET AZIMUTH DATA
         NHI   R2,X'FFF'            AND OUT UNWANTED BITS
         CR    R1,R2                COMPARE TWO AZIMUTHS
         BL    CWISE                IF R1<R2, RADAR IS GOING CWISE
         LIS   R3,0                 DIRECTION FLAG
         B     WHATIZIT              CONTINUE
CWISE    LIS   R3,15                 DIR 15LAG = CW
WHATIZIT LH    R4,INDCTR            SECTOR SCAN OR FULL CIRCLE?
         BZ    EDETECT              FULL CIRCLE
         CH    R3,CW                IS DIRECTION OF ROTATION CORRECT?
         BNE   DETECT               WRONG DIRECTION, WAIT
         OR    R3,R3                WHICH DIRECTION IS IT?
         BZ    CCW                  COUNTER CLOCKWIZE
         LH    R5,STPFLG
         THI   R5,2                 CASE 2?
         BNZ   CWCASE2              CW CASE 2
CMP1     CH    R2,BGNA              ANGLE < BGNA?
         BL    WAIT                 YES, GET READY
         B     DETECT               NO, TRY AGAIN
CWCASE2  CHI   R2,X'800'            ANGLE > 180?
         BL    CMP1                 NO,ALL OK
         SHI   R2,X'1000'           YES, SUBTRACT 360
         B     CMP1
CCW      LH    R5,STPFLG
```

```
         BNZ    CCWCASE2
CMP2     CH     R2,BGNA               ANGLE > BGNA?
         BL     DETECT                NO, TRY AGAIN
WAIT     BAL    R13,WREAD             YES, GET READY
         LR     R1,R2
         LH     R2,8(R15)             GET NEXT AZIMUTH
         NHI    R2,X'FFF'             AND OUT UNWANTED BITS
         CR     R1,R2
         BL     CW3
         LIS    R3,0
CMP6     CH     R3,CW
         BNE    DETECT
         OR     R3,R3
         BZ     CCW2
         THI    R5,2                  CASE 2?
         BNZ    WCWCASE2              YES, BRANCH
CMP3     CH     R2,BGNA               ANGLE > BGNA?
         BL     WAIT                  NO, KEEP WAITING
READ1    LH     R2,12(R15)
         NHI    R2,X'FFF'
         CHI    R2,681
         BP     ZERO
ELDET    CH     R2,ELEV1
         BM     DETECT
         CH     R2,ELEV2
         BP     DETECT
         AIS    R6,1                  INCREMENT COUNTER
         BNP    DETECT                DO IT TWICE TO BE SURE!
READ     BAL    R13,GOREAD            YES,START READING
         LH     R2,PPRANG
         NHI    R2,X'FFF'
         OR     R3,R3
         BZ     CCW1                  BRANCH IF COUNTER CLOCKWISE
         THI    R5,1                  CASE 1?
         BNZ    CWCASE1               YES, BRANCH
         CH     R2,ENDA               ANGLE > ENDA?
         BNL    DONE                  YES, ALL DONE
         B      READ                  NO, KEEP READING
CW3      LIS    R3,15
         B      CMP6
ZERO     LIS    R2,0                  59.5 DEG = 0
         B      ELDET
CWCASE1  CH     R2,BGNA               ANGLE < BGNA?
         BNL    READ                  YES, KEEP READING
         CH     R2,ENDA               NO, ARE WE DONE YET?
         BNL    DONE                  YES
         B      READ                  NO, KEEP READING
CCW1     THI    R5,1                  CHECK FOR CASE 1
         BNZ    CCWCASE1
         CH     R2,ENDA               ANGLE < ENDA?
         BL     DONE                  YES, ALL DONE
         B      READ                  NO, CONTINUE
CCWCASE1 CH     R2,BGNA               ANGLE > BGNA?
         BL     READ                  NO, KEEP READING
         CH     R2,ENDA               YES, CHECK FOR FINISHED
         BL     DONE
         B      READ
CCW2     THI    R5,2
         BNZ    WWCASE2
CMP4     CH     R2,BGNA
         BNL    WAIT                  OK
         B      READ1                 NOT YET
WWCASE2  CHI    R2,X'800'
         BNL    CMP4
         AHI    R2,X'1000'
```

```
CCWCASE2 CHI    R2,X'800'           ANGLE < 180?
         BNL    CMP2                NO, ALL OK
         AHI    R2,X'1000'          YES, ADD 360
         B      CMP2                NOW CHECK FOR BEGINNING
WCWCASE2 CHI    R2,X'800'           ANGLE > 180?
         BL     CMP3                NO,ALL OK
         SHI    R2,X'1000'          YES, SUBTRACK 360
         B      CMP3
EDETECT  BAL    R13,WREAD           READ IN NEW AZIMUTH
         LH     R2,12(R15)          READ IN ELEVATION
         NHI    R2,X'FFF'           AND OUT UNWANTED BITS
         CHI    R2,681
         BP     ZERO1
CPEV     CH     R2,ELEV1
         BL     EDETECT
         CH     R2,ELEV2
         BP     EDETECT             WITHIN RANGE?
         LH     R2,PPRANG           YES, GET AZIMUTH
         NHI    R2,X'FFF'
         CHI    R2,6
         BL     FUDGE
READ2    BAL    R13,GOREAD          START READING
         LH     R4,PPRANG
         NHI    R4,X'FFF'
NXT      OR     R3,R3
         BZ     CCLOK4              COUNTERCLOCKWISE
         CR     R4,R2
         BL     NXT2
         B      READ2
ZERO1    LIS    R2,0
         B      CPEV
FUDGE    LIS    R2,6                IF 0, MAKE IT 6
         B      READ2
NXT2     BAL    R13,GOREAD
         LH     R4,PPRANG           GET NEW AZIMUTH
         NHI    R4,X'FFF'           AND OUT UNWANTED BITS
         OR R3,R3
         BZ     CCW4
         CR     R4,R2               ANG > ENDA?
         BP     DONE                YES, FINISHED
         B      NXT2                NO,KEEP READING
CCLOK4   CR     R4,R2
         BP     NXT2
         B      READ2
CCW4     CR     R4,R2
         BL     DONE
         B      NXT2
DONE     LH     R15,PPRI
         NHI    R15,X'0FFF'
         STH    R15,PPRI
         SVC    1,OUTBLK
         LH     R0,OUTBLK+SVC1.STA
         BNZ    ERROR
         LM     0,RSAVE
$FORT
      RETURN
$ASSM
         ALIGN 4
WREAD    SVC    1,WAITREAD
         LH     R0,WAITREAD+SVC1.STA READ RETURNED STATUS
         BNZ    ERROR               IF NOT ZERO, ERROR
         BR     R13
GOREAD   SVC    1 PPRIBLK2          READ IN ONE AZIMUTH
         LM     0 RSAVE
$FORT
```

```
   10 DECOD(I)=PPRI(I)
      JSIZ=NRC/NITG
      IMX=JSIZ+1
      K=5
      DO 101 I=1,JSIZ
      REF=0
      VEL=0
      DO 20 J=1,NITG
      RAW=PPRI(K)
      K=K+1
$ASSM
       ST     R0,RSAVE
       L      R0,RAW       GET PACKED SOURCE WORD
       EXHR   R0,R0        SHIFT POWER TO LOW HALF
       NHI    R0,X'1FF'    NINE BITS
       AHM    R0,REF       INTEGRAE REFL
       L      R0,RAW       GET LOW HALF
       SRHA   R0,8         SHIFT DOPPLER TO LOW ORDER
       AHM    R0,VEL       INTEGRATE DOPPLER
       L      R0,RSAVE     RESTORE REGISTER
$FORT
   20 CONTINUE
      IF(REF.LE.0)GO TO 111
      REF=REF*.0390625-DB
      IF(REF.LT.-39) REF=REF+100
      IREF(I)=REF
      VEL=VEL*BITVEL/128.
      IVEL(I)=VEL-IVELL(I)
      IVELL(I)=VEL
      GO TO 101
  111 IREF(I)=0
      IVELL(I)=0
      IVEL(I)=0
  101 CONTINUE
$ASSM
         STM    0,RSAVE
         SVC    1,OUTBLK            OUTPUT LAST AZIMUTH
         SVC    1,WAITBLK           WAIT FOR FIRST READ TO FINISH
         SVC    1,PPRIBLK           READ NEXT AZIMUTH
         LM     0,RSAVE
$FORT
      DO 40 I=1,4
   40 DECOD2(I)=PPRI2(I)
      JSIZ=NRC/NITG
      IMX=JSIZ+1
      K=5
      DO 201 I=1,JSIZ
      REF=0
      VEL=0
      DO 400 J=1,NITG
      RAW=PPRI2(K)
      K=K+1
$ASSM
       ST     R0,RSAVE
       L      R0,RAW        GET PACKED SOURCE WORD
       EXHR   R0,R0         SHIFT POWER TO LOW HALF
       NHI    R0,X'1FF'     NINE BITS
       AHM    R0,REF        INTEGRATE REFL
       L      R0,RAW        GET LOW HALF
       SRHA   R0,8          SHIFT DOPPLER TO LOW ORDER
       AHM    R0,VEL        INTEGRATE DOPPLER
       L      R0,RSAVE      RESTORE REGISTER
$FORT
  400 CONTINUE
      IF(REF.LE.0) GO TO 411
```

```
      IF(REF.LT.-39) REF=REF+100
      IREF2(I)=REF
      VEL=VEL*BITVEL/128.
      IVEL2(I)=VEL-IVELL(I)
      IVELL(I)=VEL
      GO TO 201
  411 IREF2(I)=0
      IVEL2(I)=0
      IVELL(I)=0
  201 CONTINUE
$ASSM
         STM     0,RSAVE
         SVC     1,OUTBLK2              OUTPUT LAST AZIMUTH
         SVC     1,WAITBLK              WAIT FOR READ TO FINISH
         LH      R0,PPRIBLK2+SVC1.STA READ STATUS
         BNZ     ERROR                  IF NOT ZERO, ERROR
         LH      R0,PPRIBLK+SVC1.STA
         BNZ     ERROR
         LH      R0,OUTBLK+SVC1.STA
         BNZ     ERROR
         LH      R0,OUTBLK2+SVC1.STA
         BNZ     ERROR
         LH      R6,NAZ
         AIS     R6,2                   ADD TWO TO AZIMUTH CTR
         STH     R6,NAZ
         CHI     R6,446                 TOO MANY AZIMUTHS?
         BLR     R13                    NO, KEEP GOING
         LIS     R6,2                   YES, INDCTR=2
         STH     R6,INDCTR
         B       DONE                   QUIT
ERROR    SVC     2,ERRCODE              CONVERT ERROR CODE
         SVC     2,ERRBLOK              OUTPUT MSG TO CONSOLE
         SVC     2,PAUSE                TASK PAUSED
         LM      0,RSAVE                RESTORE FORTRAN REGISTERS
***$P2 IS FORTRAN STMT NO  2
         B       $P2                    START OVER
         ALIGN 4
PAUSE    EQU     *
         DB      0,1                    PAUSE
         ALIGN 4
ERRCODE  EQU     *
OPT      DB      0
         DB      6                      CODE 6
         DC      H'0'
         DC      A(ERCD)                DESTINATION
         ALIGN 4
ERRBLOK  EQU     *                      PRINT CONSOLE MSG
         DB      0,7                    CODE 7
         DC      H'15'                  PRINT 15 CHRS
ERCD     DC      C'     '               ERROR CODE
         DC      C'I/O ERROR'
         ALIGN 4
WAITBLK  EQU     *
         DB      X'08'                  WAIT ONLY
         DB      10                     LU 10
         DB      0,0
         DSF     5
         ALIGN 4
WAITREAD EQU     *
         DB      X'59'                  READ AND WAIT
         DB      10                     LU 10
         DB      0,0
         DC      A(PPRI)
         DC      A(PPRI)
         DSF     3
```

```
         ALIGN 4
PPRIBLK2 EQU   *
         DB    X'51'               READ
         DB    10                 LU 10
         DB    0,0
         DC    A(PPRI2)
         DC    A(PPRI2)
         DSF   3
         ALIGN 4
OUTBLK2  EQU   *
         DB    X'31'
         DB    9                  LU 9
         DB    0,0
         DC    A(DECOD2)
         DC    A(DECOD2)
         DSF   3
         ALIGN 4
PPRIBLK  EQU   *
         DB    X'51'               READ
         DB    10                 LU 10
         DB    0,0
         DC    A(PPRI)
         DC    A(PPRI)
         DSF   3
         ALIGN 4
OUTBLK   EQU   *
         DB    X'31'               WRITE
         DB    9                  LU 9
OUTST    DB    0,0
         DC    A(DECOD)
         DC    A(DECOD)
         DSF   3
$FORT
      RETURN
      END
*
```

```
*
07/19/79  16:33:21
***LISTING FOR ERT1:REALTM.FTN
$N
$ASSM
ERTRMAP PROG
$FORT
$TITL FILE   TSERMAP - - PRINT OUT DATA FIELDS-CHANGED FOR ERT READ BY CLB
      SUBROUTINE RMAP
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 NRC,VAR,STORE(10),PLACE,OLDATA
      INTEGER*4 ANC(1028),IDTIME,IZF,ITZ,IZB,IZS
     +,SRC(1028),DST(1028)
      INTEGER*2 RHO2,ZTH2,BETA2,PCT2,SIGMA2,ANG2,CHANGE
      INTEGER*2 TWENTY,ELEVEN,DAY,HOUR,MINUTE,SECOND,TP,ELEVAT,AZ
      INTEGER*2 AZIM,TC,TA,T
      INTEGER*2 Y,I,THETA1,NRC1,II,THETA,RHO,STOP,GRND
      INTEGER*2 MEAN,POWER,SIGMA,TP2,TP3,SEGNO,Q,J,ZTH,MMU
      INTEGER*2 BEGIN,SUM1,SUM2,JMIN,M,K,L,I
      INTEGER*2 Y1,ZERO,TWO,BETA
      INTEGER*4 RSAVE(16),R1SAV
      INTEGER*2 JREF(1024),JDVEL(1024)
      INTEGER*2 RE,HR,TL1,TL2,RQUANT
      REAL RNN,RRAREA
      REAL PCTMIN,AZT,ELEVAA,BGNA,AZCHK,ENDA
      INTEGER*2 FLAG,IQ,IB,IE,IM
      INTEGER*2 DBB,SLOPE
      INTEGER*4 NRCEAD
      COMMON /TLIS/ T,TA,JDAY,JHR,JMIN,JSEC,IDAY,IHR,IMIN,ISEC
      COMMON /CALR/ SRC,DST,IREF(1024)
      COMMON /SWITCH/ IC1(44),IC2(44),TC(1980),IPC1(5400),
     +   IPC2(5400),IPC3(5400),IPCNT(1980),IPTC(44),
     +   NEMB,NEMC,NAC
      COMMON/AZ2/SINA,COSA,DELTAZ,ISCANF,NEL,RI,SA
      COMMON/AZM/AZT,AZLAST,AZSTAR,NA,ELEVAA
      COMMON/REFL/RE(1025),HR(258),NCL,NID,NIDP,INCL
     X,IMX,IMN,TL1,TL2,RQUANT,IDVEL(258)
      COMMON /PNTRS/ NCMX,NVMIN,NUMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /INTL/ MHSN,MNSN,HM,FNSN
      COMMON/EXTRA/RHO,GRND,ZTH,BETA,K,RRAREA(12,24),RHO2,ZTH2,BETA2,
     1PCT2,MMU2,SIGMA2,MINUT2,ANG2(6),CHANGE
      COMMON/ZSTORE/ANC
      COMMON /ZLOOK/ IZOFF,ZARY(91)
      COMMON /ECONST/ EARTH,VMK
      COMMON /MAPPAR/ DAY,HOUR,MINUTE,SECOND,DBB,MAXV,MAXS,IOUT,SLOPE
      COMMON /CNT/ COSPHI,SINPHI,COSPH2
      COMMON /DATA1/ ECL(224),NOFST,KOFST,ICLAD,NAN1
      COMMON /DATA2/ VCL(736),MXVC,NVC1
      COMMON /DATA3/ VR(192),MXVR,NVR1
      COMMON /NVLIS/ NVARM,NCARM,NVO,ICO,IO,JO,JYR,KTL
      COMMON /FILTER/ TATRMN,AREAMN,DAZM
      EQUIVALENCE(ANC(5),JREF(2)),(ANC(517),JDVEL(2))
      DATA TWENTY/18/
      DATA TWO/2/
      DATA PCTMIN/.05/,MMU/0/,SIGMA/0/,OLDATA/0/
      DATA ZERO/0/
C
C  *******  INITIALIZE ARRAY
C
      T=0
      AZLAST=-999.
      DO 901 J=1,MXVC
  901 VCL(J)=0.
```

```
      DO 902 J=1,MAVR
  902 VR(J)=0.
      RE(1)=0
      ICC=0
C     CALL CONMSG(7,'TSERMAP')
      TL1=TL1/RQUANT
      TL2=TL2/RQUANT
      REWIND 4
      REWIND 8
      DO 3 IX=1,91
      II=IX-IZOFF
      ZX=FLOAT(II)/10.
    3 ZARY(IX)=10.**ZX
00001 CONTINUE
00750  FORMAT(1X,'ENTER PRF,0=768,1=922,2=1075,3=1229')
       RCKM=.075
       FLAG=0
333     FORMAT(I3)
       READ(7,333)OLDATA
01002 FORMAT(//'    AZIM   ELEV    RAN ',8('   REF  VEL STD'))
      REWIND 9
      K=0
10000 CONTINUE
$ASSM
R0        EQU    0
R1        EQU    1
R2        EQU    2
R3        EQU    3
R4        EQU    4
R5        EQU    5
R6        EQU    6
R7        EQU    7
R8        EQU    8
R9        EQU    9
R10       EQU    10
R11       EQU    11
R12       EQU    12
R13       EQU    13
R14       EQU    14
R15       EQU    15
          STM    R0,RSAVE
          FREZE
          COPY   SVC1.
           L      R3,ANCBLK+SVC1.SAD ANCBLK START
           AIS    R3,15              READ ANCILLARY ONLY
           ST     R3,ANCBLK+SVC1.EAD
           SVC    1,ANCBLK           THE  READ
           LH      R0,ANCBLK+SVC1.STA
           BNZ     ERROR
           LIS       R1,0
          STH    R1,STOP
          LIS    R5,0                WORKING REGISTER
          LIS    R9,10               MULTIPLICAND
          LHI    R11,100             MULTIPLICAND
          L      R0,ANC(R1)          DATA
          STBR   R0,R5               HOUR
          LIS    R10,15              MASK
          NR     R10,R5              1 HOUR
          SRLS   R5,4                10 HOUR
          NHI    R5,3
          MHR    R5,R9               (*10)
          AHR    R5,R10              TOTAL
          STH    R5,HOUR             HOURS
          EXBR   R0,R0
          STBR   R0,R5               MINUTES
          LIS    R10,15              MASK
```

```
         NR     R10,R5                 1 MINUTE
         SRLS   R5,4                   10 MINUTE
         NHI    R5,7                   MASK
         MHR    R5,R9                  (*10)
         AHR    R5,R10                 TOTAL MINUTES
         STH    R5,MINUTE
         EXHR   R0,R0
         STBR   R0,R5                  DAYS
         LIS    R10,15                 MASK
         NR     R10,R5                 1 DAY
         SRLS   R5,4                   10 DAY
         MHR    R5,R9                  (*10)
         AHR    R5,R10
         SRLS   R0,8                   100 DAYS
         NHI    R0,X'F'                MASK
         MHR    R0,R11                 (*100)
         AHR    R5,R0                  TOTAL DAYS
         STH    R5,DAY                 DAYS
         AIS    R1,4                   NEW DATA
         L      R0,ANC(R1)
         LHI   R10,256
         STH    R10,NRC1               STORE
         LHI    10,1028*4-1
         A      10,NRCBLK+SVC1.SAD
         ST     10,NRCBLK+SVC1.EAD       STORE END ADDRESS FOR READ
         ST     10,NRCEAD            STORE AWAY FOR SECOND READ
         NHI    R5,X'C0'               MASK TP
         SRLS   R5,6
         STH    R5,TP                STORE(UNFIXED)
         EXHR   R0,R0
         STBR   R0,R5                  SECONDS
         LIS    R10,15                 MASK
         NR     R10,R5                 1 SECOND
         SRLS   R5,4
         NHI    R5,7                   MASK 10 SECONDS
         MHR    R9,R5                  (*10)
         AHR    R9,R10                 TOTAL SECONDS
         STH    R9,SECOND              STORE
         AIS    R1,8                   MORE DATA
         LHL    R0,ANC(R1)             ELEVATION
         NHI    R0,X'FFF'              ANGLE
         STH    R0,ELEVAT            STORE ANGLE
         LM     R0,RSAVE
$FORT
      NRC=NRC1
       GRND=PCTMIN*NRC
      REWIND 9
17    ELEVEN=0
      TA=T
      DR=(2**TP)*RCKM
      K=K+1
      NRC1=NRC
10001 CONTINUE
$ASSM
         STM    R0,RSAVE
         SVC    1,NRCBLK             READ IN DATA
         LH     R0,NRCBLK+SVC1.STA LOAD IN STATUS
         BNZ    ERROR                IF NOT ZERO, I/O ERROR
         LIS    R1,0
         LH     R0,ANC(R1)
         BM     MINUS                  IF STOP=0
         LIS    R1,1                   STORE 1
         STH    R1,STOP
MINUS    LIS    R1,8                   GET AZIMUTH
         LH     R0,ANC(R1)             ANGLE
         NHI    R0,X'FFF'              MASK
```

```
        STH   R0,AZ                     STORE
        AIS R1,4
        LHL R0,ANC(R1)             ELEVATION
        NHI R0,X'FFF'              ANGLE
        STH R0,ELEVAT              STORE IT
$FORT
      DO 101 I=IMN,IMX
      RE(I)=JREF(I)
      IDVEL(I)=JDVEL(I)
 101  CONTINUE
00500 CONTINUE
      ICC=ICC+1
      ELEVAA=ELEVAT/11.37778
      IELSN=IFIX(ELEVAA)
      T=((DAY*24+HOUR)*60+MINUTE)*60+SECOND
      AZT=AZ/11.37778
      FLAG=FLAG+1
      A=AZT*.01743
        AZCHK=AZT
        IF(K.NE.1) GO TO 105
        BGNA=AZT
        ENDA=AZT+359.
105     CONTINUE
        IF(K.LT.180) GO TO 106
        IF(AZT.GT.180.) GO TO 106
        AZCHK=AZT+359.
106     CONTINUE
        IF(AZCHK.GT.ENDA) K=1
       SINA=SIN(A)
      COSA=COS(A)
      DELTAZ=0.0191987
      NA=K
      RE(258)=0
      IF(NA.EQ.1)NAC=1
      NAC=NAC+1
      IF(NAC.GT.1)NAC=0
      IF(NA.NE.1.OR.AZLAST.LT.-990.) GO TO 224
      IDAY=DAY
      IHR=HOUR
      IMIN=MINUTE
      ISEC=SECOND
      PHI=ELEVAA*DAZM
      COSPHI=COS(PHI)
      SINPHI=SIN(PHI)
      COSPH2=COSPHI*COSPHI*EARTH
      CALL TRACK
  224 CALL CONTOR
      AZLAST=AZT
      IF(STOP.NE.1) GO TO 17
  800 CALL STRAK
      STOP
$ASSM
        ALIGN 4
ERROR   SVC   2,ERRCODE          DECODE ERROR BITS
        SVC   2,ERRBLOK          OUTPUT ERROR MSG TO CONSOLE
        SVC   2,PAUSE
        LM    0,RSAVE            RETURN TO FORTRAN
$FORT
       IF(K.EQ.0) GO TO 10000
       IF(K.EQ.1) GO TO 10001
      GO TO 1
10003 CONTINUE
$ASSM
        ALIGN 4
ERRCODE EQU   *
        DB    0
```

```
          DB     6
          DC     H'0'
          DC     A(ERCD)
          ALIGN 4
ERRBLOK   EQU    *
          DB     0
          DB     7
          DC     H'28'
ERCD      DC     C'      '
          DC     C'I/O ERROR IN RTN RANGE'
          ALIGN 4
PAUSE     EQU    *
          DB     0,1
          ALIGN 4
ANCBLK     DB     X'59'               READ BLOCK FOR LITTLE READ
           DB     9                   LU
           DB     0,0
           DC     A(ANC)              START ADDRESS
           DC     A(ANC)              END ADDRESS
           DSF    3
NRCBLK    DB     X'59'               READ BLOCK FOR BIG READ
          DB     9                   LU
          DB     3,0                 STATUS
          DC     A(ANC)              START ADDRESS
          DC     A(ANC)              END ADDRESS
          DSF    3
$FORT
      END
*
```

APPENDIX D

POST-MISSION ANALYSIS VERSION

(note BLOCK DATA, CONTOR, PEAKD,
TRACK, ATRAK, BTRAK, COMPAR,
RESOLV, COMBIN and STRAK are
identical in both versions)

```
$ASSM
          SCRAT                      TSEDATA (1)
          SQUEZ 3
TSEDATA   PROG
$FORT
$TITL FILE TSEDATA - DATA INPUT SUBROUTINE FOR ETSE
      SUBROUTINE PPRDSC(ANG)
      IMPLICIT INTEGER*2 (A-Z)
      INTEGER*4 PPRI(1028),PPRI2(1028),RSAVE(16)
       INTEGER*4 OUT(260),OUT2(260)
      INTEGER*4 PPRANG
      INTEGER*2 ANG(6)
      EQUIVALENCE (PPRANG,PPRI(3)),(PP,PPRI(2))
       EQUIVALENCE (OUT(1),PPRI(1)),(OUT2(1),PPRI2(1))
      COMMON /SECTOR/INDCTR
      COMMON /ZSTORE/PPRI
      COMMON /RUNSUM/PPRI2
       AVEN=3
      CALL CONMSG(6,'PPRDSC')
2     REWIND 9
      NAZ=0
      ELEV1=ANG(4)*11.37778
      ELEV2=ANG(5)*11.37778
      IF(INDCTR.EQ.2)INDCTR = 0
      IF (INDCTR .EQ. 0)GO TO 1
      CW = ANG(3) - ANG(2)
      IF (CW .GT. 180) CW = CW - 360
      IF (CW .LT. (-180)) CW = 360 + CW
      IF (CW .LT. 0) CW = 0
      IF (CW .GT.0) CW = 15
      BGNA = 11.37778 * ANG(2)
      ENDA = 11 37778 * ANG(3)
      BGNA=MOD(BGNA,4096)
      ENDA=MOD(ENDA,4096)
      IF(ENDA.GT.4086)ENDA=0
      STPFLG = 0
      IF (CW .GT. 0 .AND. ENDA .LT. BGNA) STPFLG = 1
      IF (CW .EQ. 0 .AND. ENDA .GT. BGNA) STPFLG = 1
      IF(BGNA.GT.10 .AND. BGNA.LT.4086)GO TO 1
      BGNA=0
      STPFLG=2
 1    CONTINUE
10000 CONTINUE
$ASSM
          FREZE
          CROSS
          COPY  SVC1.
R0        EQU   0
R1        EQU   1
R2        EQU   2
R3        EQU   3
R4        EQU   4
R5        EQU   5
R6        EQU   6
R14       EQU   14
R15       EQU   15
R13       EQU   13
R7        EQU  7
R8        EQU  8
R9        EQU 9
R10       EQU  10
R11       EQU  11
R12       EQU  12
```

```
         STM    0,RSAVE
         L      R4,WAITREAD+SVC1.SAD
         AIS    R4,15
         ST     R4,WAITREAD+SVC1.EAD
         BAL    R13,WREAD
         LIS    R1,4
         L      R0,PPRI(R1)
         NHI    R0,3
         AIS    R0,1
         SLLS   R0,8
         AIS    R0,4
         SLLS   R0,2
         SIS    R0,1
         L      R4,PPRIBLK+SVC1.SAD GET BEGINNING ADDRESS
         AR     R4,R0             COMPUTE END ADDRESS
         ST     R4,PPRIBLK+SVC1.EAD STORE IN EAD
         ST     R4,WAITREAD+SVC1.EAD SAME NUMBER
*
*    LETS ONLY 256 WORDS OF VIDEO OUT
*
       SI   R4,2048
         ST     R4,OUTBLK+SVC1.EAD
         L      R4,PPRIBLK2+SVC1.SAD GET NEXT BEGINNING ADDR
         AR     R4,R0             COMPUTE END ADDRESS
         ST     R4,PPRIBLK2+SVC1.EAD STORE IN SVC BLOCK
*
*     LETS ONLY 256 WORDS OF VIDEO OUT
*
       SI   R4,2048
         ST     R4,OUTBLK2+SVC1.EAD
         LDAI   R15,PPRI
         LDAI   R14,PPRI2
         LCS    R6,1                SET R6 TO -1 FOR COUNTER
DETECT   BAL    R13,WREAD           READ IN AN AZIMUTH
         LH     R1,8(R15)           LOAD IN AZIMUTH DATA
         NHI    R1,X'FFF'           AND OUT UNWANTED BITS
         BAL    R13,WREAD           GET ANOTHER AZIMUTH
         LH     R2,8(R15)           GET AZIMUTH DATA
         NHI    R2,X'FFF'           AND OUT UNWANTED BITS
         CR     R1,R2               COMPARE TWO AZIMUTHS
         BL     CWISE               IF R1<R2, RADAR IS GOING CWISE
         LIS    R3,0                DIRECTION FLAG
         B      WHATIZIT             CONTINUE
CWISE    LIS    R3,15                DIR 15LAG = CW
WHATIZIT LH     R4,INDCTR           SECTOR SCAN OR FULL CIRCLE?
         BZ     EDETECT             FULL CIRCLE
         CH     R3,CW               IS DIRECTION OF ROTATION CORRECT?
         BNE    DETECT              WRONG DIRECTION, WAIT
         OR     R3,R3               WHICH DIRECTION IS IT?
         BZ     CCW                 COUNTER CLOCKWIZE
         LH     R5,STPFLG
         THI    R5,2                CASE 2?
         BNZ    CWCASE2             CW CASE 2
CMP1     CH     R2,BGNA             ANGLE < BGNA?
         BL     WAIT                YES, GET READY
         B      DETECT              NO, TRY AGAIN
CWCASE2  CHI    R2,X'800'           ANGLE > 180?
         BL     CMP1                NO,ALL OK
         SHI    R2,X'1000'          YES, SUBTRACT 360
         B      CMP1
CCW      LH     R5,STPFLG
         THI    R5,2                CHECK FOR CASE 2
         BNZ    CCWCASE2
CMP2     CH     R2,BGNA             ANGLE > BGNA?
         BL     DETECT              NO, TRY AGAIN
WAIT     BAL    R13,WREAD           YES, GET READY
```

```
         LH      R2,8(R15)         GET NEXT AZIMUTH
         NHI     R2,X'FFF'         AND OUT UNWANTED BITS
         CR      R1,R2
         BL      CW3
         LIS     R3,0
CMP6     CH      R3,CW
         BNE     DETECT
         OR      R3,R3
         BZ      CCW2
         THI     R5,2              CASE 2?
         BNZ     WCWCASE2          YES, BRANCH
CMP3     CH      R2,BGNA           ANGLE > BGNA?
         BL      WAIT              NO, KEEP WAITING
READ1    LH      R2,12(R15)
         NHI     R2,X'FFF'
         CHI     R2,681
         BP      ZERO
ELDET    CH      R2,ELEV1
         BM      DETECT
         CH      R2,ELEV2
         BP      DETECT
         AIS     R6,1              INCREMENT COUNTER
         BNP     DETECT            DO IT TWICE TO BE SURE!
READ     BAL     R13,GOREAD        YES,START READING
         LH      R2,PPRANG
         NHI     R2,X'FFF'
         OR      R3,R3
         BZ      CCW1              BRANCH IF COUNTER CLOCKWISE
         THI     R5,1              CASE 1?
         BNZ     CWCASE1           YES, BRANCH
*    WANT SEVERAL PPIS
       B   READ
         CH      R2,ENDA           ANGLE > ENDA?
         BNL     DONE              YES, ALL DONE
         B       READ              NO, KEEP READING
CW3      LIS     R3,15
         B       CMP6
ZERO     LIS     R2,0              59.5 DEG = 0
         B       ELDET
CWCASE1  CH      R2,BGNA           ANGLE < BGNA?
         BNL     READ              YES, KEEP READING
*   WANT SEVERAL PPIS
       B    READ
         CH      R2,ENDA           NO, ARE WE DONE YET?
         BNL     DONE              YES
         B       READ              NO, KEEP READING
CCW1     THI     R5,1              CHECK FOR CASE 1
         BNZ     CCWCASE1
         CH      R2,ENDA           ANGLE < ENDA?
         BL      DONE              YES, ALL DONE
         B       READ              NO, CONTINUE
CCWCASE1 CH      R2,BGNA           ANGLE > BGNA?
         BL      READ              NO, KEEP READING
         CH      R2,ENDA           YES, CHECK FOR FINISHED
         BL      DONE
         B       READ
CCW2     THI     R5,2
         BNZ     WWCASE2
CMP4     CH      R2,BGNA
         BNL     WAIT              OK
         B       READ1             NOT YET
WWCASE2  CHI     R2,X'800'
         BNL     CMP4
         AHI     R2,X'1000'
         B       CMP4
```

```
CCWCASE2 CHI    R2,X'800'            ANGLE < 180?
         BNL    CMP2                 NO, ALL OK
         AHI    R2,X'1000'           YES, ADD 360
         B      CMP2                 NOW CHECK FOR BEGINNING
WCWCASE2 CHI    R2,X'800'            ANGLE > 180?
         BL     CMP3                 NO,ALL OK
         SHI    R2,X'1000'           YES, SUBTRACK 360
         B      CMP3
EDETECT  BAL    R13,WREAD            READ IN NEW AZIMUTH
         LH     R2,12(R15)           READ IN ELEVATION
         NHI    R2,X'FFF'            AND OUT UNWANTED BITS
         CHI    R2,681
         BP     ZERO1
CPEV     CH     R2,ELEV1
         BL     EDETECT
         CH     R2,ELEV2
         BP     EDETECT              WITHIN RANGE?
         LH     R2,PPRANG            YES, GET AZIMUTH
         NHI    R2,X'FFF'
         CHI    R2,6
         BL     FUDGE
READ2    BAL    R13,GOREAD           START READING
         LH     R4,PPRANG
         NHI    R4,X'FFF'
NXT      OR     R3,R3
         BZ     CCLOK4               COUNTERCLOCKWISE
         CR     R4,R2
         BL     NXT2
         B      READ2
ZERO1    LIS    R2,0
         B      CPEV
FUDGE    LIS    R2,6                 IF 0, MAKE IT 6
         B      READ2
NXT2     BAL    R13,GOREAD
         LH     R4,PPRANG            GET NEW AZIMUTH
         NHI    R4,X'FFF'            AND OUT UNWANTED BITS
         OR R3,R3
         BZ     CCW4
*   WANT SEVERAL PPIS
       B    NXT2
         CR     R4,R2                ANG > ENDA?
         BP     DONE                 YES, FINISHED
         B      NXT2                 NO,KEEP READING
CCLOK4   CR     R4,R2
         BP     NXT2
         B      READ2
CCW4     CR     R4,R2
         BL     DONE
         B      NXT2
DONE     LH     R15,PPRI
         NHI    R15,X'0FFF'
         STH    R15,PPRI
         SVC    1,OUTBLK
         LH     R0,OUTBLK+SVC1.STA
         BNZ    ERROR
         LM     0,RSAVE
$FORT
      RETURN
$ASSM
         ALIGN 4
WREAD    SVC    1,WAITREAD
         LH     R0,WAITREAD+SVC1.STA READ RETURNED STATUS
         BNZ    ERROR                IF NOT ZERO, ERROR
         BR     R13
GOREAD   SVC    1,PPRIBLK2           READ IN ONE AZIMUTH
```

```
        LIS    R12,12        COUNTER FOR MAIN BUFFER
      LHI    R4,16      COUNTER FOR NEW BUFFER
AVE     LIS    R8,0        ZERO POWER ACCUMULATOR
        LIS    R10,0       ZERO VELOCITY ACCUMLTR
        LIS    R1,3        AVERAGING COUNTER
AVER    LHL    R7,PPRI(R4)     LOAD POWER HALFWORD
        NHI    R7,X'1FF'       GET ONLY THE POWER
        AR     R8,R7           ADD TO ACCUMLTR
        AIS    R4,2            INCREMENT ARRAY POINTER
        LHL    R7,PPRI(R4)     LOAD VEL/VAR HALFWORD
        NHI    R7,X'FFFF'
        AR     R10,R7          INCREMENT FOR NEXT HALFWORD
        AIS    R4,2            INCREMENT FOR NEXT HALFWORD
        SIS    R1,1            SUBTRACT FOR INNER LOOP
        BNZ    AVER            BRANCH IF NOT 3 ADDED UP
        DH     R8,AVEN         DIVIDE POWER
        NHI    R9,X'1FF'
        SLA  R9,16           SHIFT POWER TO LEFT
       SRA   R10,8
        DH     R10,AVEN        DIVIDE VELOCITY
       SLA  R11,8
        NHI    R11,X'FFFF'       GET ONLY LOWER HALFWORD
        AR     R9,R11          PUT TOGETHER
        AIS    R12,4             INCREMENT NEW ARRAY POINTER
        ST     R9,PPRI(R12)  PUT INTO ARRAY
        CLHI   R4,3072
        BP     AVEEND
        B      AVE
*
AVEEND    SVC   1,OUTBLK    OUTPUT LA T AZIMUTH
          SVC   1,WAITBLK          WAIT FOR FIRST READ TO FINISH
          SVC   1,PPRIBLK          READ NEXT AZIMUTH
          SVC   1,OUTBLK2          OUTPUT LAST AZIMUTH
          SVC   1,WAITBLK          WAIT FOR READ TO FINISH
          LH    R0,PPRIBLK2+SVC1.STA READ STATUS
          BNZ   ERROR              IF NOT ZERO, ERROR
          LH    R0,PPRIBLK+SVC1.STA
          BNZ   ERROR
          LH    R0,OUTBLK+SVC1.STA
          BNZ   ERROR
          LH    R0,OUTBLK2+SVC1.STA
          BNZ   ERROR
          LH    R6,NAZ
          AIS   R6,2               ADD TWO TO AZIMUTH CTR
          STH   R6,NAZ
          CHI   R6,1760             TOO MANY AZIMUTHS?
          BLR   R13                NO, KEEP GOING
          LIS   R6,2               YES, INDCTR=2
          STH   R6,INDCTR
          B     DONE               QUIT
ERROR     SVC   2,ERRCODE          CONVERT ERROR CODE
          SVC   2,ERRBLOK          OUTPUT MSG TO CONSOLE
          SVC   2,PAUSE            TASK PAUSED.
          LM    0,RSAVE            RESTORE FORTRAN REGISTERS
***$P2 IS FORTRAN STMT NO. 2
          B     $P2                START OVER
          ALIGN 4
PAUSE     EQU   *
          DB    0,1                PAUSE
          ALIGN 4
ERRCODE   EQU   *
OPT       DB    0
          DB    6                  CODE 6
          DC    H'0'
          DC    A(ERCD)            DESTINATION
          ALIGN 4
```

```
         ALIGN 4
ERRBLOK  EQU    *
         DB     0,7                     PRINT CONSOLE MSG
         DC     H'15'                   CODE 7
ERCD     DC     C'                      PRINT 15 CHRS
         DC     C'I/O ERROR'            ERROR CODE
         ALIGN 4
WAITBLK  EQU    *
         DB     X'08'                   WAIT ONLY
         DB     10                      LU 10
         DB     0,0
         DSF    5
         ALIGN 4
WAITREAD EQU    *
         DB     X'59'                   READ AND WAIT
         DB     10                      LU 10
         DB     0,0
         DC     A(PPRI)
         DC     A(PPRI)
         DSF    3
         ALIGN 4
PPRIBLK2 EQU    *
         DB     X'51'                   READ
         DB     10                      LU 10
         DB     0,0
         DC     A(PPRI2)
         DC     A(PPRI2)
         DSF    3
         ALIGN 4
OUTBLK2  EQU    *
         DB     X'31'
         DB     9                       LU 9
         DB     0,0
         DC     A(OUT2)
         DC     A(OUT2)
         DSF    3
         ALIGN 4
PPRIBLK  EQU    *
         DB     X'51'                   READ
         DB     10                      LU 10
         DB     0,0
         DC     A(PPRI)
         DC     A(PPRI)
         DSF    3
         ALIGN 4
OUTBLK   EQU    *
         DB     X'31'                   WRITE
         DB     9                       LU 9
OUTST    DB     0,0
         DC     A(OUT)
         DC     A(OUT)
         DSF    3
$FORT
      RETURN
      END
```

```
*
07/19/79  12:49:01
***LISTING FOR ERT1:CRANE.FTN
$N
$ASSM
ERTRMAP PROG
$FORT
$TITL FILE   TSERMAP - - PRINT OUT DATA FIELDS-CHANGED FOR ERT READ BY CLB
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 NRC,VAR,STORE(10),PLACE,OLDATA
      INTEGER*4 ZEE(1024),ANC(1028),IDTIME,IZF,ITZ,IZB,IZS
      INTEGER*2 TWENTY,ELEVEN,DAY,HOUR,MINUTE,SECOND,TP,ELEVAT,AZ
      INTEGER*2 AZIM,TC,TA,T
      INTEGER*2 Y,I,THETA1,NRC1,II,THETA,RHO,STOP,GRND
      INTEGER*2 MEAN,POWER,SIGMA,TP2,TP3,SEGNO,Q,J,ZTH,MMU
      INTEGER*2 BEGIN,SUM1,SUM2,JMIN,M,K,L,I
      INTEGER*2 Y1,ZERO,TWO,BETA
      INTEGER*4 RSAVE(16),R1SAV
      INTEGER*2 IREF(1024),IVEL(1024),IVAR(1024),IVELL(256)
     +   ,JDVEL(256)
      INTEGER*2 RE,HR,TL1,TL2,RQUANT
      REAL RNN,RRAREA,AXX
      REAL PCTMIN,AZT,ELEVAA,BGNA,AZCHK,ENDA
      INTEGER*2 FLAG,IQ,IB,IE,IM
      INTEGER*2IOUT,DBB,MAXV,MAXS,SLOPE
      INTEGER*4 NRCEAD
      COMMON /TLIS/ T,TA,JDAY,JHR,JMIN,JSEC,IDAY,IHR,IMIN,ISEC
      COMMON /CONST/ VMISW,DIV,VMAG,VMISWM,ZDIV,ADIV,A1,A2,A3,B1,B2,HDIV
      COMMON /SWITCH/ IC1(44),IC2(44),TC(1980),IPC1(5400),
     +   IPC2(5400),IPC3(5400),IPCNT(1980),IPTC(44),
     +   NEMB,NEMC,NAC
      COMMON/AZ2/SINA,COSA,DELTAZ,ISCANF,NEL,RI,SA
      COMMON/AZM/AZT,AZLAST,AZSTAR,NA,ELEVAA
      COMMON/REFL/RE(1025),HR(258),NCL,NID,NIDP,INCL
     X,IMX,IMN,TL1,TL2,RQUANT,IDVEL(258)
      COMMON /PNTHS/ NCMX,NVMIN,NUMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /INTL/ MHSN,MNSN,HM,FNSN
      COMMON/EXTRA/RHO,GRND,ZTH,BETA,K,RRAREA(288),RHO2(16),AXX(288),
     1IPRFM
      COMMON/ZSTORE/ANC
      COMMON /ZLOOK/ IZOFF,ZARY(91)
      COMMON /ECONST/ EARTH,VMK
      COMMON /MAPPAR/ DAY,HOUR,MINUTE,SECOND,DBB,MAXV,MAXS,IOUT,SLOPE
      COMMON /CNT/ COSPHI,SINPHI,COSPH2
      COMMON /DATA1/ ECL(224),NOFST,KOFST,ICLAD,NAN1
      COMMON /DATA2/ VCL(736),MXVC,NVC1
      COMMON /DATA3/ VR(192),MXVR,NVR1
      COMMON /NVLIS/ NVARM,NCARM,NVO,ICO,IO,JO,JYR,KTL
      COMMON /FILTER/ TATRMN,AREAMN,DAZM
      EQUIVALENCE(RE(2),IREF(1)),(IDVEL(2),JDVEL(1))
      EQUIVALENCE(ANC(5),ZEE(1))
      DATA TWENTY/18/
      DATA TWO/2/
      DATA PCTMIN/.05/,MMU/0/,SIGMA/0/,OLDATA/0/
      DATA ZERO/0/
C
C  *******  INITIALIZE ARRAY
C
      T=0
      AZLAST=-999.
      DO 901 J=1,MXVC
  901 VCL(J)=0.
      DO 902 J=1,MXVR
```

```
      VMISWM=(VMISW-1.)/DIV
      RE(1)=0
      JDVEL(1)=0
      JDVEL(256)=0
      ICC=0
C     CALL CONMSG(7,'TSERMAP')
      TL1=TL1/RQUANT
      TL2=TL2/RQUANT
      REWIND 4
      REWIND 8
      DO 2 I=1,1024
      IREF(I)=0
      IVEL(I)=0
    2 IVAR(I)=0
      DO 3 IX=1,91
      II=IX-IZOFF
      ZX=FLOAT(II)/10.
    3 ZARY(IX)=10.**ZX
00001 CONTINUE
      DB=65.28
  750 FORMAT(1X,'ENTER PRF,0=768,1=922,2=1075,3=1229')
      RCKM=.075
      FLAG=0
      READ(7,333)IOUT,DBB,MAXV,MAXS,SLOPE
333   FORMAT(I3)
      READ(7,333)OLDATA
01002 FORMAT(//'   AZIM  ELEV   RAN ',8('  REF VEL STD'))
      BITVEL=MAXV
      BITVAR=MAXS
      REWIND 9
      K=0
10000 CONTINUE
$ASSM
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
         STM   R0,RSAVE
         FREZE
         COPY  SVC1.
          L     R3,ANCBLK+SVC1.SAD ANCBLK START
          AIS   R3,15              READ ANCILLARY ONLY
          ST    R3,ANCBLK+SVC1.EAD
          SVC   1,ANCBLK           THE  READ
          LH     R0,ANCBLK+SVC1.STA
          BNZ    ERROR
          LIS     R1,0
         STH   R1,STOP
         LIS   R5,0                WORKING REGISTER
         LIS   R9,10               MULTIPLICAND
         LHI   R11,100             MULTIPLICAND
         L     R0,ANC(R1)          DATA
         STBR  R0,R5               HOUR
```

```
         LIS   R10,15               MASK
         NR    R10,R5               1 HOUR
         SRLS  R5,4                 10 HOUR
         NHI   R5,3
         MHR   R5,R9                (*10)
         AHR   R5,R10               TOTAL
         STH   R5,HOUR              HOURS
         EXBR  R0,R0
         STBR  R0,R5                MINUTES
         LIS   R10,15               MASK
         NR    R10,R5               1 MINUTE
         SRLS  R5,4                 10 MINUTE
         NHI   R5,7                 MASK
         MHR   R5,R9                (*10)
         AHR   R5,R10               TOTAL MINUTES
         STH   R5,MINUTE
         EXHR  R0,R0
         STBR  R0,R5                DAYS
         LIS   R10,15               MASK
         NR    R10,R5               1 DAY
         SRLS  R5,4                 10 DAY
         MHR   R5,R9                (*10)
         AHR   R5,R10
         SRLS  R0,8                 100 DAYS
         NHI   R0,X'F'              MASK
         MHR   R0,R11               (*100)
         AHR   R5,R0                TOTAL DAYS
         STH   R5,DAY               DAYS
         AIS   R1,4                 NEW DATA
         L     R0,ANC(R1)
         LIS   R5,0                 JUST TO BE SURE
         STBR  R0,R5                NRC AND TP
         LIS   R10,3                MASK
         NR    R10,R5               NRC
         AIS   R10,1
         LHI   R12,256
         MHR   R10,R12              NRC=(NRC+1)*256
       LHI   R10,256
         STH   R10,NRC1             STORE
         SLHLS 10,2               MULTIPLY BY 4 FOR BYTE COUNT
         AIS   10,15
         A     10,NRCBLK+SVC1.SAD
         ST    10,NRCBLK+SVC1.EAD     STORE END ADDRESS FOR READ
         ST    10,NRCEAD          STORE AWAY FOR SECOND READ
         NHI   R5,X'C0'             MASK TP
         SRLS  R5,6
         STH   R5,TP              STORE(UNFIXED)
         EXHR  R0,R0
         STBR  R0,R5                SECONDS
         LIS   R10,15               MASK
         NR    R10,R5               1 SECOND
         SRLS  R5,4
         NHI   R5,7                 MASK 10 SECONDS
         MHR   R9,R5                (*10)
         AHR   R9,R10               TOTAL SECONDS
         STH   R9,SECOND            STORE
         AIS   R1,8                 MORE DATA
         LHL   R0,ANC(R1)           ELEVATION
         NHI   R0,X'FFF'            ANGLE
         STH   R0,ELEVAT          STORE ANGLE
         LM    R0,RSAVE
$FORT
      NRC=NRC1
      GRND=PCTMIN*NRC
      REWIND 9
17    ELEVEN=0
```

```
      ... .
      DR=(2**TP)*RCKM
      K=K+1
      NRC1=NRC
10001 CONTINUE
$ASSM
         STM    R0,RSAVE
         SVC    1,NRCBLK            READ IN DATA
         LH     R0,NRCBLK+SVC1 STA LOAD IN STATUS
         BNZ    ERROR               IF NOT ZERO, I/O ERROR
         LIS    R1,0
         LH     R0,ANC(R1)
         BM     MINUS                IF STOP=0
         LIS    R1,1                 STORE 1
         STH    R1,STOP
MINUS    LIS    R1,8                 GET AZIMUTH
         LH     R0,ANC(R1)           ANGLE
         NHI    R0,X'FFF'            MASK
         STH    R0,AZ                STORE
        AIS R1,4
        LHL R0,ANC(R1)          ELEVATION
        NHI R0,X'FFF'           ANGLE
        STH R0,ELEVAT           STORE IT
         LHL    R8,NRC1              MAX. NO. OF CELLS
         LHI    R3,32767             MAX. HALFWORD
         LDAI   R15,GRND
         LHL    R1,0(R15)            CELL COUNTER
         LIS    R2,0                 SUM ZR
         LIS    R4,7                 FOR
         LIS    R5,0                 SHIFTING
         LIS    R6,0                 SUM ZR2/128
         LDAI   R15,ZTH
         LHL    R7,0(R15)            MIN. POWER
         LIS    R9,0                 CELLS PER SEG.
         LDAI   R15,RHO
         LHL    R10,0(R15)           MIN. SEGMENT SIZE
         STH    R10,PLACE           INITIALIZE PLACE
         SIS    R10,1
         LHL    R11,MMU              MINIMUM MEAN
         LHL    R12,SIGMA            MINIMUM VARIANCE
         B      POW
*
*            MAIN LOOP
*
LOOP     AR     R1,R10
NLOOP    AIS    R1,1                 NEXT CELL
         CR     R1,R8                IF DONE,
         BNL    REALLY                 LEAVE
POW      LR     R15,R1
         SLLS   R15,2
         L      R5,ZEE(R15)          RAW DATA
         EXHR   R0,R5                POWER
          LH     R2,OLDATA
          BNZ    NO9
         NHI    R0,X'1FF'
          STH    R0,IREF(R1)       STORE REFLECTIVITY
CONT     EXHR   R15,R5              PUT MEAN IN TOP OF R15
         SRA    R15,24              SHIFT DOWN 24 WITH SIGN EXT.
         BNMS   POSITIVE             ABSOLUTE VALUE
         XHI    R15,X'FFFF'          COMPLEMENT
         AIS    R15,1                TWO'S COMPLEMENT
          STH    R15,IVEL(R1)         STORE VELOCITY
POSITIVE NHI    R5,X'FF'             VARIANCE
          STH    R5,IVAR(R1)          STORE VARIANCE
          B       NLOOP
NO9      NHI    R0,X'FF'
```

```
         B      CONT
*
*               DATA STORING ROUTINE
*
REALLY   LM     R0,RSAVE
$FORT
        DO 101 I=1,256
        IF (IREF(I).LE.0) GO TO 111
        IREF(I)=IREF(I)*100./256.-DB
        IF(IREF(I).LT.-39) IREF(I)=IREF(I)+100
        IVEL(I)=IVEL(I)*BITVEL/128.
        IDVEL(I)=IVEL(I)-IVELL(I)
        IVELL(I)=IVEL(I)
C       F4WORD=IVAR(I)*BITVAR/256.
C      IVAR(I) = SQRT(F4WORD)
        GO TO 101
00111  IREF(I)=0
        IVEL(I)=0
        IVAR(I)=0
  101   CONTINUE
00500  CONTINUE
       ICC=ICC+1
       ELEVAA=ELEVAT/11.37778
       IELSN=IFIX(ELEVAA)
       T=((DAY*24+HOUR)*60+MINUTE)*60+SECOND
       AZT=AZ/11.37778
        FLAG=FLAG+1
       A=AZT*.01743
         AZCHK=AZT
         IF(K.NE.1) GO TO 105
         BGNA=AZT
         ENDA=AZT+359.
105      CONTINUE
         IF(K.LT.180) GO TO 106
         IF(AZT.GT.180.) GO TO 106
         AZCHK=AZT+359.
106      CONTINUE
         IF(AZCHK.GT.ENDA) K=1
        SINA=SIN(A)
       COSA=COS(A)
       DELTAZ=0.0191987
       NA=K
       RE(258)=0
       IF(NA.EQ.1)NAC=1
       NAC=NAC+1
       IF(NAC.GT.1)NAC=0
       IF(NA.NE.1.OR.AZLAST.LT.-990.) GO TO 224
       IDAY=DAY
       IHR=HOUR
       IMIN=MINUTE
       ISEC=SECOND
       PHI=ELEVAA*DAZM
       COSPHI=COS(PHI)
       SINPHI=SIN(PHI)
       COSPH2=COSPHI*COSPHI*EARTH
       CALL TRACK
  223  IF(NSCAN.EQ.4) GO TO 800
  224  CALL CONTOR
       AZLAST=AZT
       IF(STOP.NE.1) GO TO 17
  800  CALL STRAK
       STOP
$ASSM
        ALIGN 4
ERROR   SVC    2,ERRCODE          DECODE ERROR BITS
```

```
         SVC    2,ERRBLOK               OUTPUT ERROR MSG TO CONSOLE
         SVC    2,PAUSE
         LM     0,RSAVE                 RETURN TO FORTRAN
$FORT
       IF(K.EQ.0) GO TO 10000
       IF(K.EQ.1) GO TO 10001
      GO TO 1
10003 CONTINUE
$ASSM
         ALIGN 4
ERRCODE  EQU    *
         DB     0
         DB     6
         DC     H'0'
         DC     A(ERCD)
         ALIGN 4
ERRBLOK  EQU    *
         DB     0
         DB     7
         DC     H'28'
ERCD     DC     C'        '
         DC     C'I/O ERROR IN RTN RANGE'
         ALIGN 4
PAUSE    EQU    *
         DB     0,1
         ALIGN 4
ANCBLK    DB     X'59'                   READ BLOCK FOR LITTLE READ
          DB     9                       LU
          DB     0,0
          DC     A(ANC)                  START ADDRESS
          DC     A(ANC)                  END ADDRESS
          DSF    3
NRCBLK   DB     X'59'                   READ BLOCK FOR BIG READ
         DB     9                       LU
         DB     0,0                     STATUS
         DC     A(ANC)                  START ADDRESS
         DC     A(ANC)                  END ADDRESS
         DSF    3
$FORT
      END
*
```

```
*
07/19/79  12:51:51
***LISTING FOR ERT1:INPARM.FTN
$N
      BLOCK DATA
C     ****************************************************************
C     FOR PROGRAM EXTRAD   ERT NO. 162
C     VERSION 4.0    LEVEL 780301
C     JHW    IBM370
C     ****************************************************************
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 W,TL1,TL2,HR,RQUANT,T,TM
C     ----------------------------------------------------------------
      COMMON /CDRAYS/ IC(32,10),C(32,9),ID(32,10),D(32,9),IM,JM
      COMMON /CONST/ VMISW,DIV,VMAG,VMISWM,ZDIV,ADIV,A1,A2,A3,B1,B2,HDIV
      COMMON /VPARM/ VX,VY
      COMMON /PNTRS/ NCMX,NVMIN,NUMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /ECONST/ EARTH,VMK
      COMMON /NVLIS/ NVARM,NCARM,NVO,ICO,IO,JO,JYR,KTL
      COMMON /ZLOOK/ IZOFF,ZARY(91)
      COMMON /INTL/ MHSN,MNSN,HM,FNSN
      COMMON /DVAL/ DELA
      COMMON /CNTRS/ IATR(5),MATR
      COMMON /FLGS/ ITYPE,PR1,PR2,PRIN2
      COMMON /TMAX/ TM
      COMMON /AZM/ AZMUTH,AZLAST,AZSTAR,NA,ELEVAT
      COMMON /DATA1/ ECL(224),NOFST,KOFST,ICLAD,NAN1
      COMMON /DATA2/ VCL(736),MXVC,NVC1
      COMMON /DATA3/ VR(192),MXVR,NVR1
      COMMON/FILTER/TATRMN,AREAMN,DAZM
      COMMON /PWORK/ KMAX,T(80),JMXDB,JMAX,IAMAX,IR,JR,IMXJMX
      COMMON /AZ2/SINA,COSA,DELTAZ,ISCANF,NEL,RI,SA
      COMMON /FIXED/ NPA,IEMAX,NFC,IEM,JEM
      COMMON /PRSTOR / NUP,TATR(1400),NUMAX,IACT(70),
     +   IDC(22),IPRNG(34),KAMAX,MXTR
      COMMON/REI L/ W(1025),HR(258),NCL,NID,NIDP,INCL,
     X IMX,IMN,TL1,TL2,RQUANT,IDVEL(258)
      COMMON/THRESH/LDV
      LOGICAL PR1,PR2,PRIN2
      DATA ECL/224*0.0/,VCL/736*0.0/,VR/192*0.0/
      DATA TL1/30/,TL2/60/
      DATA NOFST/16/,KOFST/7/,ICLAD/112/
      DATA  DAZM/0.0174533/,ISCANF/0/
      DATA RI/0../,SA/900./,RQUANT/1/
      DATA LDV/3./,MXVR/192/,NVR1/6/,MXVC/736/,NVC1/23/
      DATA NPA/2/,IEMAX/22/,NFC/2/
      DATA KMAX/45/,JMXDB/80/,JMAX/34/,IAMAX/2700/,IR/15/,JR/45/,
     +      IMXJMX/60/,NCL/258/,KAMAX/990/
      DATA NID/250/,NUP/4/,NIDP/70/,NUMAX/20/,MXTR/1400/
      DATA AREAMN/0.4/,IMX/257/,IMN/2/
      DATA IZOFF/10/,DELA/0.0087/,FNSN/0.009/
      DATA IATR/9,8,5,0,0/
      DATA ITYPE/1/,PR1/.TRUE./,PR2/.FALSE./,PRIN2/.FALSE./
      DATA NVARM/32/,MNSN/5/,MHSN/7/,HM/6.5/
      DATA TM/0/,NCARM/16/,VMK/1.  E-3/
      DATA IM/32/,JM/9/,NVO/0/,ICO/0/,IO/0/,JO/0/
      DATA VMISW/5./,DIV/.2/,VMAG/.01/,ZDIV/.1/,HDIV/.5/
      DATA ADIV/.04/,A1/.4/,A2/.3/,A3/.3/,B1/.7/,B2/.3/
      DATA VX/0./,VY/0./,IESNL/0/,IELSN/0/
      DATA EARTH/6.4857 E-5/,NT/0/,NSCAN/0/,NVSCN/1/
      END
*
```

AD-A081 061   ENVIRONMENTAL RESEARCH AND TECHNOLOGY INC CONCORD MA   F/G 17/9
AUTOMATIC WEATHER RADAR ECHO ASSESSMENT AND TRACKING.(U)
MAR 79 R K CRANE   F19628-78-C-0076
UNCLASSIFIED   AFGL-TR-79-0248   NL

2 OF 2
AD
A08106

END
DATE
FILMED
3-80
DDC

```
*
07/19/79  12:52:43
***LISTING FOR ERT1:CONTOR.FTN
$N
      SUBROUTINE CONTOR
C     ***************************************************************
C     JHW   AFGL SUDBURY RADAR SUBROUTINE
C     VERSION 4.1   LEVEL 781117
C     FIND EVENTS ALONG SINGLE RADIAL
C     ***************************************************************
      IMPLICIT INTEGER*2 (I-N)
      INTEGER *2 W,TL1,TL2,HR,RQUANT,TC
     +  ,T,IC21(22),IC22(22)
      COMMON /SWITCH/ IC1(44),IC2(44),TC(1980),IPC1(5400),
     +   IPC2(5400),IPC3(5400),IPCNT(1980),IPTC(44),
     +   NEMB,NEMC,NAC
      COMMON/AZM/ AZMUTH,AZLAST,AZSTAR,NA,ELEVAT
      COMMON /AZ2/SINA,COSA,DELTAZ,ISCANF,NEL,RI,SA
      COMMON/FILTER/TATRMN,AREAMN,DAZM
      COMMON /FIXED/ NPA,IEMAX,NFC,IEM,JEM
      COMMON /PRSTOR / NUP,TATR(1400),NUMAX,IACT(70),
     +       IDC(22),IPRNG(34),KAMAX,MXTR
      COMMON/REFL/ W(1025),HR(258),NCL,NID,NIDP,INCL
     X,IMX,IMN,TL1,TL2,RQUANT,IDVEL(258)
      COMMON /PWORK/ KMAX,T(80),JMXDB,JMAX,IAMAX,IR,JR,IMXJMX
      COMMON /THRESH/LDV
      COMMON /CONPK/ NCEL,NMR
C     ---------------------------------------------------------------
      DATA RPD/.017453/
C     ---------------------------------------------------------------
C     CALL CONMSG(6,'CONTOR')
C     IFLAG=1
C     IF(IFLAG.EQ.1)WRITE(3,7)(W(IX),IX=1,102)
C     FORMAT(1X,20I5)
      IF (NA.NE.1) GO TO 61
C
C              INITIALIZE.
C
      TLS=TL1*RQUANT
      TATRMN=AREAMN*1.E06/SA
      NEMC=IEMAX
      NCEL=1
      JEM=0
61    CONTINUE
      DO 101 K=1,IEMAX
      IDC(K)=0
101   CONTINUE
      DO 102 J=1,JMAX
      IPRNG(J)=0
102   CONTINUE
      NEMB=NEMC
      NEMC=NAC*IEMAX
      NEM1=NEMC+1
      IEM=0
      IEM2=0
       IP=0
      IPB=0
C
C              FIND EVENTS
C
      DO 281 I=2,NCL 258
      IF(W(I).GT.TLS) GO TO 2311
      W(I)=0
```

```
 2311 IF(RQUANT.GT.1)W(I)=W(I)/RQUANT
      IF (W(I).GT.TL1) GO TO 131
      GO TO 241
131   IF (W(I-1).LE.TL1) GO TO 141
      GO TO 151
141   IEM=IEM+1
      IEA=IEM+NEMC
      IF(IEM.LE.IEMAX)GO TO 1411
      WRITE(3,1412)IEMAX,K
1412  FORMAT(1X,39HEVENT COUNTER EXCEEDED MAX VALUE, IMAX=,I6,5X,I4)
      IEM=IEMAX
 1411 IC1(IEA)=I-1
      IC2(IEA)=0
C
C                PEAK DETECTION, LOCATE AND COUNT PEAKS.
C
  151 IF (W(I)-W(I-1)) 171,181,161
161   IPB=I-1
      GO TO 181
171   IF (IPB.EQ.0) GO TO 181
      IP=IP+1
      IF(IP.LE.JMAX)GO TO 1711
      WRITE(3,1913)IP,IEVENT
 1913 FORMAT(1X,17HN PEAKS EXCEEDED, 2I6)
      IP=JMAX
      GO TO 181
1711  IPRNG(IP)=(I+IPB)/2
      IPB=0
181   CONTINUE
      GO TO 282
  241 IF (W(I-1).LE.TL1) GO TO 281
      IC2(IEA)=I-1
C
C                KEEP COUNT OF PEAKS WITH EVENT.
C
      IF (IPB.EQ.0) GO TO 251
      IP=IP+1
      IF(IP.LE.JMAX)GO TO 242
      WRITE(6,1913)IP,IEM
      IP=JMAX
      GO TO 243
242   IPRNG(IP)=(I+IPB)/2
243   IPB=0
251   IDC(IEM)=IP
  282 IF(W(I).LE.TL2) GO TO 2412
      IF(W(I-1).GT.TL2) GO TO 281
      IEM2=IEM2+1
      IF(IEM2.GT.IEMAX)IEM2=IEMAX
      IC21(IEM2)=I-1
      IC22(IEM2)=0
      GO TO 281
 2412 IF(W(I-1).LE.TL2) GO TO 281
      IC22(IEM2)=I-1
281   CONTINUE
      CALL PEAKD
C
C                STORE PRESENT PARAMETERS IN PREVIOUS PARAMETERS.
C
C     IF(IEL.NE.1) GO TO 802
C     WRITE(6,1) AZMUTH,TL2,IEM2,(IC21(J),IC22(J),J=1,IEM2)
C   1 FORMAT(F8.2,2I8,/11(2I6,3X)/11(2I6,3X))
C 801 WRITE(8) AZMUTH,TL1,IEM,(IC1(J),IC2(J),J=NEM1,IEA)
      JEM=IEM
      RETURN
      END
```

```
*
07/19/79  12:54:07
***LISTING FOR ERT1:PEAKD.FTN
$N
      SUBROUTINE PEAKD
C     ******************************************************************
C     VERSION 5.0   LEVEL 780616
C     JHW     AVCO     IBM360
C     DETERMINES PEAK VALUES AND THEIR ATTRIBUTES.
C     ******************************************************************
      IMPLICIT INTEGER*2 (I-N)
      REAL UP(6),TATR(1400),BUF(8)
      INTEGER*4 RSAVE(16)
      INTEGER *2 TCVL,TBVL,TCVLB,TCVM,TATC,TATM
      INTEGER *2  HB,IACT(70),IDC(22),
     +   IPCRNG(34),TM
      INTEGER *2 W,RQUANT
      INTEGER*2 TL2,T,TC
      COMMON /DATA1/ ECL(224),NOFST,KOFST,ICLAD,NAN1
      COMMON /SWITCH/ IC1(44),IC2(44),TC(1980),IPC1(5400),
     +   IPC2(5400),IPC3(5400),IPCNT(1980),IPTC(44),
     +   NEMB,NEMC,NAC
      COMMON /PWORK/ KMAX,T(80),JMXDB,JMAX,IAMAX,IR,JR,IMXJMX
      COMMON/REFL/ W(1025),HB(258),NCL,NID,NIDP,INCL,
     X IMX,IMN,TM,TL2,RQUANT,IDVEL(258)
      COMMON /FIXED/NPA,IEMAX,NFC,IEM,JEM
      COMMON/AZM/ AZMUTH,AZLAST,AZSTAR,NA,ELEVAT
      COMMON /AZ2/SAZ,CAZ,DAZ,ISCANF,NEL,RI,SA
      COMMON/FILTER/TATRMN,AREAMN,DAZM
      COMMON /THRESH/ LDB
      COMMON /PRSTOR/ NUP,TATR(1400),NUMAX,IACT(70),
     +     IDC(22),IPCRNG(34),KAMAX,MXTR
      COMMON /CONPK/ NCELL,NMX
      DATA DPR/57.296/,A/0.1/
C
C              IEM IS NO.OF EVENTS IN C RADIAL.
C              INITIALIZE AND GENERATE HC ARRAY
C
      NBADR=NCADR
      NCADR=IAMAX*NAC
      NBKA=NCKA
      NCKA=KAMAX*NAC
      NAX=NA
      ITY=0
      IF(NA.NE.1) GO TO 2109
      LM=6
      LMDP=LM*NIDP
      NBADR=IAMAX*NAC
      NBKA=KAMAX*NAC
      NAN=0
      NAN1=0
      LMM=LM-1
      IDX=LM+1
      NIMN=1
      FQUANT=RQUANT
      NCLM=NCL-1
      LDBM=LDB-1
      NUMP=2+LM*LDB
      LDX=NUMP-1
      NPDP=LDX*NIDP
      ID2=1+(LDB-1)*LM
      DO 17 I=1,KOFST
   17 UP(I)=0.
```

```
      DO 18 I=1,LIMT
   18 ECL(I)=0.
      DO 19 I=1,NCL
   19 IDVEL(I)=0
C
      NMX=1
      DO 2107 I=1,NIDP
 2107 IACT(I)=0
      DO 2108 J=1,MXTR
2108  TATR(J)=0.
2109  NGM=0
      DO 23 K=1,KMAX
      KAD=K+NCKA
   23 IPCNT(KAD)=0
C
 1044 IF(IEM.LE.0)GO TO 952
      DO 951 IE=1,IEM
      IEA=IE+NEMC
      KIE=(IE-1)*KMAX+NCKA
      KIEM=KIE-KMAX
      IPTC(IEA)=0
      ICEST=IC1(IEA)
      ICESP=IC2(IEA)
      IF(IE.EQ.1)GO TO 232
      DO 233 K=1,KMAX
      KA=K+KIE
      KB=K+KIEM
233   IPCNT(KA)=IPCNT(KB)
  232 IPL=0
      IF(IE.GT.1)IPL=IDC(IE-1)
      IP=IDC(IE)
      IF(IP.LE.IPL)GO TO 951
      IPL=IPL+1
      JE1=0
      JE2=0
C              FIND B EVENTS ASSOCIATED WITH C EVENTS.
C              JEM IS NO. OF EVENTS IN PREVIOUS RADIAL.
C
      IF(JEM .EQ. 0) GO TO 41
      DO 31 JE=1,JEM
      JEB=JE+NEMB
      IF(IC2(JEB) .LT. ICEST) GO TO 31
      IF(IC1(JEB).GT. ICESP) GO TO 41
      JE2=JE
      IF (JE1.EQ.0) JE1=JE
31    CONTINUE
C              FIND THRESHOLDS FOR IE EVENT
C
41    DO 51 J=1,JMXDB
51    T(J)=0
      NTHRES=1
      DO 71 L=IPL,IP
      IF(L.GT.JMAX)GO TO 71
      IR1=IPCRNG(L)
      IF(IR1.LT.ICEST)GO TO 71
      IF(IR1.GT.ICESP)GO TO 712
      DO 711 K=1,LDB
      IU=W(IR1)
      IT=IU-TM-K+1
      IF(IT.LE.0)GO TO 711
      IF(IT.GT.JMXDB)IT=JMXDB
      IF(T(IT).EQ.0)NTHRES=NTHRES+1
      T(IT)=1
711   CONTINUE
71    CONTINUE
```

```
      IF(NTHRES.GT.KMAX)WRITE(6,7878)NA,NTHRES,KMAX,IE
 7878 FORMAT(2X,33HNUMBER OF THRESHOLDS EXCEEDS KMAX,4I10)
      IF(NTHRES.GT.KMAX)IPSRT=NTHRES-KMAX
      IPT=1
      DO 91 L=1,JMXDB
      IF (T(L).LE.0) GO TO 91
      KA=IPT+KIE
      TC(KA)=L+TM-1
      IPSRT=IPSRT-1
      IF(IPSRT.GT.0)GO TO 91
      IPT=IPT+1
91    CONTINUE
      IPT=IPT-1
      IF(IPT.GT.JR)WRITE(6,7879)NA,IPT,JR,IE
 7879 FORMAT(2X,31HNUMBER OF THRESHOLDS EXCEEDS JR,4I10)
      IF(IPT.GE.JR)IPT=JR
      IPTC(IEA)=IPT
      IF(IPT.LE.0)GO TO 951
C               LOOP ON RANGE IN IE EVENT TO FIND CONTOUR
C
      IBGN=ICEST+1
      IND=ICESP+1
      DO 161 I=IBGN,IND
C
C               LOOP ON THRESHOLD
C
      DO 131 K=1,IPT
      IU=W(I)
      KA=K+KIE
      IF (IU.GT.TC(KA)) GO TO 111
      GO TO 141
  111 IU=W(I-1)
      IF (IU.LE.TC(KA)) GO TO 121
      GO TO 131
C
C               START RANGE FOR SEGMENT (CONTOUR)
C
121   IPCNT(KA)=IPCNT(KA)+1
      IF(IPCNT(KA).LE.IMXJMX)GO TO 1211
      WRITE(6,1212)ITY,K,IE
1212  FORMAT(2X,30HNUMBER OF SEGMENTS EXCEEDS IMX,3I10)
      IPCNT(KA)=IMXJMX
1211  IPE=IPCNT(KA)
      IREG=I-1
      IADDR=IPE+(K-1)*IMXJMX+NCADR
      IPC1(IADDR)=IREG
      IPC3(IADDR)=0
131   CONTINUE
      GO TO 161
C
C               END RANGE FOR SEGMENT
C
141   DO 151 KL=K,IPT
      IF (W(I-1).EQ.-999) GO TO 161
      IU=W(I-1)
      KA=KL+KIE
      IF (IU.LE.TC(KA)) GO TO 161
      IPE=IPCNT(KA)
      IREG=I-1
      IEQL=IPE+(KL-1)*IMXJMX+NCADR
      IPC2(IEQL)=IREG
151   CONTINUE
161   CONTINUE
C
C               ASSOCIATE CELLS   LOOP ON THRESHOLD HIGHEST TO LOWEST
```

```
940    DO 941 LC=1,IPT
       KC=IPT-LC+1
       KA=KC+KIE
       KZ=KC+KIEM
       IF(KC.LE.0)GO TO 941
       TCVL=TC(KA)
       TCVM=TCVL+1
       TCVLB=TCVL+LDB
       NPC=IPCNT(KA)
       NPL=0
       IF(IE.GT.1)NPL=IPCNT(KZ)
       IF(NPC.LE.NPL)GO TO 941
       NPL=NPL+1
C               LOOP ON SEGMENTS
       DO 931 IPE=NPL,NPC
       JADDR=IPE+(KC-1)*IMXJMX+NCADR
       IHBM=IPC1(JADDR)
       IHB=IHBM+1
       IHD=IPC2(JADDR)
       K=KC+1
       KY=KA+1
       KX=KZ+1
       NPK=0
       TATM=0
       LPE=IPCNT(KY)
       LPL=0
       IF(IE.GT.1)LPL=IPCNT(KX)
       LPL=LPL+1
       IF(LPE.LT.LPL.OR.K.GT.IPT)GO TO 193
C
C           LOOP SEGMENTS, NEXT HIGHER THRESHOLD
C
  192 DO 191 L=LPL,LPE
       IADDR=L+(K-1)*IMXJMX+NCADR
       IF(IPC2(IADDR).LT.IHBM)GO TO 191
       IF(IPC1(IADDR).GT.IHD)GO TO 193
       NPCEL=IPC3(IADDR)
       IF(NPCEL.LE.0)GO TO 1911
       TATC=TATR(NPCEL)
       IF(TATC.LT.TATM) GO TO 231
       TATM=TATC
       NPK=NPCEL
  231 IF(TATC.LT.0)TATC=-TATC
C
C               NPCEL IS FOR NEXT HIGHER (ENCLOSED) THRESHOLD ON C RADIAL
C
       IF(TATC.GT.TCVLB)GO TO 932
191    CONTINUE
       GO TO 193
932    NPK=-NPCEL
       GO TO 193
1911   NPK=-(NIDF+1)
C
C               ASSOCIATE CELLS ON B RADIAL, TOP DOWN
C
193    MPK=0
       IF(NA.EQ.1)GO TO 361
       TATM=0
       IF (JE2.EQ.0) GO TO 371
       DO 261 JE=JE1,JE2
       JEB=JE+NEMB
       ITATM=0
       IF(IC2(JEB).LT.IHBM) GO TO 261
       IF(IC1(JEB).GT.IHD) GO TO 3661
C
```

```
C
271     IPB=IPTC(JEB)
        IF(IPB.LE.0)GO TO 261
        DO 291 LB=1,IPB
        KB=IPB-LB+1
        KBB=(JE-1)*KMAX+NBKA
        KBA=KB+KBB
        KBC=KB+KBB-KMAX
        TBVL=TC(KBA)
        JEQL=TBVL+1
        NP2=IPCNT(KBA)
        NP1=0
        IF(JE.GT.1)NP1=IPCNT(KBC)
        IF(NP2.LE.NP1)GO TO 291
        NP1=NP1+1
        DO 281  JPE=NP1,NP2
        IADDR=JPE+(KB-1)*IMXJMX+NBADR
        IF(IPC2(IADDR).LT.IHBM)GO TO 281
        IF(IPC1(IADDR).GT.IHD)GO TO 2911
        LPCEL=IPC3(IADDR)
        IF(LPCEL.LE.0)GO TO 281
        IF(TCVL .LE. TBVL) GO TO 282
        IEQL=TATR(LPCEL)
        IF(JEQL.LT.IEQL)GO TO 281
  282   TATC=TATR(LPCEL)
        IF(TATC.LT.TATM) GO TO 281
        TATM=TATC
        ITATM=TATM
        MPK=LPCEL
        KBM=KB
        JBM=JE
281     CONTINUE
2911    IF(ITATM.NE.0) GO TO 261
291     CONTINUE
261     CONTINUE
3661    IF(MPK.EQ.0)GO TO 371
        IF(TATM.GT.TCVLB)MPK=-MPK
        GO TO 421
371     DO 194 I=IHB,IHD
        IF(HB(I).EQ.-999)GO TO 194
        IF(IABS(HB(I)).LE.TC(KA) )GO TO 194
        IF(NPK.EQ.0)GO TO 931
        IF(NPK.GT.0)GO TO 366
        GO TO 3662
194     CONTINUE
C
C                 HAVE B COMPARE WITHIN RANGE
C
361     CONTINUE
        IF( NPK.EQ.0)GO TO 631
C
C                 MPK=0.AND.NPK=0 - NO COMPARE
C                 MPK=0.AND.NPK.NE.0 - NO B COMPARE
C                 NPK=0.AND.MPK.NE.0 - B COMPARE
C                 HIGHEST THIS RADIAL
C
        IF(NPK.LE.0.OR.NPK.GT.NMX)GO TO 3662
C
C                 NO PRIOR RADIAL FOR COMPARISON, INCREMENT NPCEL
C
        NPCEL=NPK
359     INDX=TATR(NPCEL)-TCVM
        IF(INDX.GE.LDB.OR.INDX.LE.0)GO TO 366
392     IN=INDX*LM+1
        INX=IDX+INDX*LM
```

```
      IEQL=TATR(JN)
      IF(IEQL NE 0 OR NA EQ 1)GO TO 3921
      IEQL=NPCEL+(IN-LMM-1)*NIDP
      JEQL=TFTR(IEQL)
      IF(JEQL LE 0)GO TO 366
      MPC=NPCEL
      NPCEL=JEQL
      IF(MPC EQ NPCEL OR NPCEL GT NMX)GO TO 366
      GO TO 359
 3921 IPC3(JADDR)=NPCEL
      JN1=JN+NIDP
      JN2=JN1+NIDP
      JN3=JN2+NIDP
      JN4=JN3+NIDP
      JN5=JN4+NIDP
      IF(TATR(JN1) EQ -999 )GO TO 419
      IST=IHB
      ISP=IHD
      DO 411 I=IST,ISP
      R=RI+SA*(FLOAT(I-1)-.5)
      IU=W(I)
      RU=R*FLOAT(IU)*DAZ
      TATR(JN1)=TATR(JN1)+DAZ*R
      TATR(JN2)=TATR(JN2)+ RU
      TATR(JN3)=TATR(JN3)+SAZ*R*RU
      TATR(JN4)=TATR(JN4)+CAZ*R*RU
      TATR(JN5)=TATR(JN5)+IDVEL(I)
411   CONTINUE
 419  KN=NPCEL+(INX-1)*NIDP
      TATR(KN)=NA
      KNN=NPCEL+NPDP
      TATR(KNN)=IE
      IF(IST EQ 2 OR ISP EQ IMX)TATR(JN1)=-999
      GO TO 366
3662  NPCEL=-NPK
  366 IF(NPCEL GT NMX OR NPCEL LE 0)GO TO 931
      INDX=TATR(NPCEL)-TCVM
C
C                COMBINE LPCEL WITH NPCEL AT THIS LEVEL
C         COMBINE BY SETTING AREA AS POINTER AND IDX TO NA = 0
C
      IF(LPE LT LPL OR K GT IPT)GO TO 931
      DO 365 L=LPL,LPE
      IADDR=L+(K-1)*IMXJMX
      IF(IPC2(IADDR) LT IHBM)GO TO 365
      IF(IPC1(IADDR) GT IHD)GO TO 931
      LPCEL=IPC (IADDR)
      IF(LPCEL LE 0 OR LPCEL GT NMX)GO TO 365
      LNX=LPCEL+LMDP
      IF(TATR(LNX) EQ 0 )GO TO 365
      IF(NPCEL EQ LPCEL)GO TO 365
      INDX=TATR(LPCEL)-TCVM
      IF(INDX GE LDB)GO TO 365
      IF(INDX LE 0)INDX=0
      IND=LPCEL+(INDX+1)*LMDP
      IF(TATR(IND) EQ 0 )GO TO 365
      IND=INDX+1
      IPG=0
      DO 3663 J=IND,LDB
      IN=(J-1)*LM
      IEQL=LPCEL+J*LMDP
      JEQL=TATR(IEQL)
      IF(JEQL EQ NA)IPG=IPG+1
      DO 3663 I=1,LM
      IEQL=LPCEL+(I+IN)*NIDP
```

```
      IF(IPG.EQ.0.OR.IE.LE.1)GO TO 3664
      DO 3665 I=1,IE
      IA=I+NEMC
      IPTT=IPTC(IA)
      IF(IPTT.LE.0)GO TO 3665
      DO 3666 KT=1,IPTT
      KTI=(I-1)*KMAX+NCKA
      KTA=KT+KTI
      KTB=KT+KTI-KMAX
      NPCT=IPCNT(KTA)
      IEQL=TC(KTA)+1
      NPCL=0
      IF(I.GT.1)NPCL=IPCNT(KTB)
      IF(NPCT.LE.NPCL)GO TO 3666
      NPCL=NPCL+1
      DO 3667 LP=NPCL,NPCT
      IADDR=LP+(KT-1)*IMXJMX+NCADR
      IF(LPCEL.NE.IPC3(IADDR))GO TO 3667
      INDXT=TATR(NPCEL)-IEQL
      IF(INDXT.LT.LDB)GO TO 3668
3669  IPC3(IADDR)=0
      GO TO 3667
3668  IF(IMDX.GE.LDB)GO TO 3669
      IPC3(IADDR)=NPCEL
3667  CONTINUE
3666  CONTINUE
3665  CONTINUE
      IPG=0
3664  IF(IMDX.GE.LDB)GO TO 365
      IACT(LPCEL)=-NPCEL
      IEQL=LPCEL+(INDX*LM+1)*NIDP
      TATR(IEQL)=NPCEL
      IF(INDX.NE.0)GO TO 365
      IACT(LPCEL)=-NIDP-1
      LN=LPCEL+NIDP
      TATR(LN)=0
365   CONTINUE
      GO TO 931
C
C             COMBINE NPCEL AND LPCEL, PEAK VALUES EQUAL
C
C
C              COMBINE WITH E RADIAL CELLS
C
421   IF(MPK.LE.0)GO TO 422
      IF(NPK.LT.0)GO TO 3662
      NGM=0
      LPCEL=MPK
      LNX=LPCEL+LMDP
      IEQL=TATR(LNX)
      KBMA=KBM+(JBM-1)*KMAX+NBKA
      IF(IEQL.EQ.NA.AND.NPK.EQ.0.AND.TCVL.GT.
     +   TC(KBMA)) GO TO 485
      INDX=TATR(LPCEL)-TCVM
      IMDX=INDX
      IF(NPK.GT.0)IMDX=TATR(NPK)-TCVM
      IF(IMDX.LE.INDX)GO TO 4212
      NGM=1
      NPCEL=NPK
      IND=INDX
      INDX=IMDX
      IMDX=IND
      GO TO 4213
4212  IF(INDX.LT.0)GO TO 481
      NPCEL=LPCEL
```

```
C
C                 COMBINE WITH B - RADIAL, C-LEVEL LOWER
C
4213  IF(INDX.GE.LDB)GO TO 4221
      IN=INDX*LM
      IN1=(IN+1)*NIDP
      ILN=(IN+LM)*NIDP
      IEQL=NPCEL+ILN
512   IF(TATR(IEQL).NE.0.)GO TO 5311
      IEQL=NPCEL+IN1
      JEQL=TATR(IEQL)
      IF(JEQL.LE.0.AND.NGM.EQ.0)GO TO 4221
      IF(NGM.NE.1)GO TO 5312
5314  IM=IMDX*LM
      IF(IM.LT.0) GO TO 5311
      IF(LPCEL.LE.0.OR.LPCEL.GT.NMX) GO TO 422
      IEQL=LPCEL+(IM+LM)*NIDP
      IF(TATR(IEQL).NE.0.)GO TO 5311
      IEQL=LPCEL+(IM+1)*NIDP
      IF(TATR(IEQL).GT.0.)GO TO 5313
      LPCEL=NPCEL
      GO TO 4221
5313  LPCEL=TATR(IEQL)
      IF(LPCEL.EQ.NPCEL.OR.LPCEL.GT.NMX)GO TO 4221
      IMDX=TATR(LPCEL)-TCVM
      GO TO 5314
5312  IEQL=NPCEL+IN1
      NPCEL=TATR(IEQL)
      IF(NPCEL.LE.0.OR.NPCEL.GT.NMX)GO TO 4221
      INDX=TATR(NPCEL)-TCVM
      GO TO 4213
5311  IPC3(JADDR)=NPCEL
      IEQL=NPCEL+IN1
      IF(TATR(IEQL).EQ.-999.) GO TO 8012
      IST=IHB
      ISP=IHD
      IEQL2=IEQL+NIDP
      IEQL3=IEQL2+NIDP
      IEQL4=IEQL3+NIDP
      IEQL5=IEQL4+NIDP
      DO 531 I=IST,ISP
      IU=W(I)
      R=RI+SA*(FLOAT(I-1)-.5)
      RU=R*FLOAT(IU)*DAZ
      TATR(IEQL)=TATR(IEQL)+DAZ*R
      TATR(IEQL2)=TATR(IEQL2)+RU
      TATR(IEQL3)=TATR(IEQL3)+SAZ*R*RU
      TATR(IEQL4)=TATR(IEQL4)+CAZ*R*RU
      TATR(IEQL5)=TATR(IEQL5)+IDVEL(I)
531   CONTINUE
 8012 IEQL=NPCEL+ILN
      TATR(IEQL)=NA
      NMP=NPCEL+NPDP
      TATR(NMP)=IE
      IEQL=NPCEL+IN1
      IF(IST.EQ.2.OR.ISP.EQ.IMX)TATR(IEQL)=-999.
      LPCEL=NPCEL
      GO TO 4221
C
C                 COMBINE WITH B-RADIAL, C-LEVEL HIGHER
C
C                 IF FIRST COMBINE, AREA=0, IF SECOND OR HIGHER, AREA=-1.
C                 TEST AREA TO ESTABLISH NEW NUMBERS
C
  481 INDX=-INDX
      IND=NUMP-1
```

```
      INDP=LDB
      INS=2
      IPG=0
      TATR(LPCEL)=TCVM
      LMP=LPCEL+NPDP
      TATR(LMP)=IE
      IF(INDX.GE.LDB)GO TO 482
      IND=LDB-INDX
      DO 4832 I=INDX,LDBM
      IEQL=LPCEL+(I+1)*LMDP
      JEQL=TATR(IEQL)
      IF(JEQL.EQ.NA)IPG=IPG+1
4832  CONTINUE
      DO 483 I=1,IND
      DO 483 J=1,LM
      IN=LPCEL+(J+(LDB-I)*LM)+NIDP
      IM=LPCEL+(J+(IND-I)*LM)+NIDP
483   TATR(IN)=TATR(IM)
      IND=INDX*LM+1
      INDP=INDX
482   DO 4835 I=1,LDB
      IEQL=LPCEL+I*LMDP
      JEQL=TATR(IEQL)
      IF(JEQL.EQ.NA)IPG=IPG+1
4835  CONTINUE
      DO 484 I=INS,IND
      IN=LPCEL+(I-1)*NIDP
484   TATR(IN)=0
      DO 4841 I=1,INDP
      IEQL=LPCEL+I*LMDP
4841  TATR(IEQL)=NA
      IF(IPG.EQ.0.OR.IE.LE.1)GO TO 488
      DO 4831 I=1,IE
      IA=I+NEMC
      IPTT=IPTC(IA)
      IF(IPTT.LE.0)GO TO 4831
      DO 4833 KT=1,IPTT
      KTP=(I-1)*KMAX+NCKA
      KTA=KT+KTP
      KTB=KT+KTP-KMAX
      NPCT=IPCNT(KTA)
      IEQL=TC(KTA)+1
      NPCL=0
      IF(I.GT.1)NPCL=IPCNT(KTB)
      IF(NPCT.LE.NPCL)GO TO 4833
      NPCL=NPCL+1
      DO 4834 LP=NPCL,NPCT
      KADDR=LP+(KT-1)*IMXJMX+NCADR
      IF(LPCEL.NE.IPC3(KADDR))GO TO 4834
      INDXT=TATR(LPCEL)-IEQL
      IF(INDXT.LT.LDB)GO TO 4834
      IPC3(KADDR)=0
4834  CONTINUE
4833  CONTINUE
4831  CONTINUE
      IPG=0
488   IN=0
      IF(LPCEL.LE.0.OR.LPCEL.GT.NMX)GO TO 931
      LND=LPCEL+LMDP
      TATR(LND)=NA
      IPC3(JADDR)=LPCEL
      NPCEL=LPCEL
      NGM=0
      GO TO 512
485   DO 486 I=NIMN,NIDP
      IF(IACT(I).EQ.0)GO TO 487
```

```
      WRITE(6,644)
      GO TO 931
  487 LPCEL=I
      NIMN=I+1
      IACT(I)=1
      IF(NIMN.GT.NIDP) NIMN=NIDP
      IF(NMX.LT.NIMN) NMX=NIMN
      TATR(LPCEL)=TCVM
      LNP=LPCEL+NPDP
      TATR(LNP)=IE
      GO TO 488
  422 LPCEL=MPK
      IF(LPCEL.LT.0)LPCEL=-LPCEL
 4221 IF(LPCEL.GT.NMX.OR.LPCEL.LE.0)GO TO 3662
      IMDX=TATR(LPCEL)-TCVM
      IF(IMDX.LT.0)GO TO 632
      DO 441 JE=JE1,JE2
      JEB=JE+NEMB
      IF(IC2(JEB).LT.IHBM) GO TO 441
      IF(IC1(JEB).GT.IHD) GO TO 632
      IPB=IPTC(JEB)
      IF(IPB.LE.0)GO TO 441
      DO 471 LB=1,IPB
      KB=IPB-LB+1
      KBJ=(JE-1)*KMAX+NBKA
      KBA=JE+KBJ
      KBC=JE+KBJ-KMAX
      IF(TC(KBA).NE.TCVL) GO TO 471
      MPB=IPCNT(KBA)
      MPL=0
      IF(JE.GT.1)MPL=IPCNT(KBC)
      IF(MPB.LE.MPL)GO TO 471
      MPL=MPL+1
      DO 461 JPE=MPL,MPB
      IADDR=JPE+(KB-1)*IMXJMX+NBADR
      IF(IPC2(IADDR).LT.IHBM)GO TO 461
      IF(IPC1(IADDR).GT.IHD)GO TO 471
      NPCEL=IPC3(IADDR)
      IF(NPCEL.LE.0.OR.NPCEL.GT.NMX)GO TO 461
      IF(LPCEL.EQ.NPCEL)GO TO 461
C
C        COMBINE AT TB=TC LEVEL
C
502   INDX=TATR(NPCEL)-TCVM
      IF(INDX.GE.LDB)GO TO 461
      IF(INDX.LT.0)GO TO 8511
      IF(IMDX.LT.LDB)GO TO 861
851   ND=INDX*LM+1
      DO 852 I=1,LM
      IEQL=NPCEL+(ND+I-1)*NIDP
852   TATR(IEQL)=0.
      IPC3(IADDR)=0
      GO TO 461
8511  IPC3(IADDR)=0
      DO 8512 J=1,LDX
      JN=NPCEL+J*NIDP
8512  TATR(JN)=0.
      IACT(NPCEL)=-(NIDP+1)
      GO TO 461
861   LD=IMDX*LM+1
      ND=INDX*LM+1
      LDA=LD*NIDP
      NDA=ND*NIDP
      IEQL=LPCEL+(LD+LMM)*NIDP
      JEQL=NPCEL+(ND+LMM)*NIDP
```

```
      TATRL=TATR(IEQL)
      TATRN=TATR(JEQL)
      IF(TATRL.NE.0..AND.TATRN.NE.0.)GO TO 8911
      JEQL=LPCEL+LDA
      TATRJ=TATR(JEQL)
      IF(TATRL.EQ.0.  .AND.TATRJ.LE.0.)GO TO 851
      IF(TATRL.GT.0.)GO TO 8912
      LPCEL=TATRJ
      IF(LPCEL.LE.0.OR.LPCEL.GT.NMX)GO TO 461
      GO TO 4221
 8912 IEQL=NPCEL+NDA
      TATRJ=TATR(IEQL)
      IF(TATRN.EQ.0..AND.TATRJ.LE.0.)GO TO 8913
      NPCEL=TATRJ
      IF(NPCEL.LE.0.OR.NPCEL.GT.NMX.OR.NPCEL.EQ.LPCEL)GO TO 461
      IPC3(IADDR)=NPCEL
      GO TO 502
8913  DO 8914 I=1,LM
      IEQL=LPCEL+(LD+I-1)*NIDP
8914  TATR(IEQL)=0.
      IPC3(IADDR)=0
      GO TO 4221
 8911 IBNDRY=0
      IEQL=LPCEL+LDA
      JEQL=NPCEL+NDA
      IF(TATR(IEQL).EQ.-999..OR.TATR(JEQL).EQ.-999.)
     X IBNDRY=1
      DO 891 I=1,LMM
      IEQL=LPCEL+(LD+I-1)*NIDP
      JEQL=NPCEL+(ND+I-1)*NIDP
      IF(IBNDRY.EQ.0)TATR(IEQL)=TATR(JEQL)+TATR(IEQL)
      TATR(JEQL)=0.
891   CONTINUE
      IEQL=LPCEL+LDA
      IF(IBNDRY.EQ.1)TATR(IEQL)=-999.
      IEQL=NPCEL+(ND+LMM)*NIDP
      TATR(IEQL)=0.
      IEQL=NPCEL+NDA
      TATR(IEQL)=LPCEL
      IACT(NPCEL)=-LPCEL
      IPC3(IADDR)=LPCEL
461   CONTINUE
471   CONTINUE
441   CONTINUE
632   IF(NPK.LE.0)GO TO 3662
      NPCEL=LPCEL
      GO TO 366
C
C               UNASSOCIATED
C
631   DO 642 J=NIMN,NIDP
      IF(IACT(J).EQ.0)GO TO 643
642   CONTINUE
      WRITE(6,644)
644   FORMAT(5X,15H TOO MANY CELLS)
      GO TO 931
643   NPCEL=J
      NIMN=J
      IF(NIMN.GT.NIDP)NIMN=NIDP
      IF(NMX.LT.NIMN)NMX=NIMN
      IACT(J)=1
      IPC3(JADDR)=NPCEL
      DO 671 I=1,NUMP
      NMP=NPCEL+(I-1)*NIDP
      TATR(NMP)=0.0
671   CONTINUE
```

```
      NMN=NPCEL+NPDP
      TATR(NMN)=IE
      IST=IHB
      ISP=IHD
      NP2=NPCEL+NIDP
      NP3=NP2+NIDP
      NP4=NP3+NIDP
      NP5=NP4+NIDP
      NP6=NP5+NIDP
      DO 621 I=IST,ISP
      R=RI+SA*(FLOAT(I-1)-.5)
      IU=W(I)
      RU=R*FLOAT(IU)*DAZ
      TATR(NP2)=DAZ*R+TATR(NP2)
      TATR(NP3)=RU+TATR(NP3)
      TATR(NP4)=SAZ*R*RU+TATR(NP4)
      TATR(NP5)=TATR(NP5)+CAZ*R*RU
      TATR(NP6)=TATR(NP6)+IDVEL(I)
621   CONTINUE
      NIX=NPCEL+LMDP
      TATR(NIX)=NA
      IF(IST.EQ.2.OR.ISP.EQ.IMX)TATR(NP2)=-999.
931   CONTINUE
941   CONTINUE
951   CONTINUE
C
C        CLEAN UP TATR AND IC ARRAYS  -  REMOVE IC POINTER
C        TO DELETED ARRAYS
C
7010  DO 9512 I=1,NMX
      IF(IACT(I).EQ.0)GO TO 9512
      IF(IACT(I).GE.0)GO TO 9611
      DO 9613 IE=1,IEM
      IEA=IE+NEMC
      KIE=(IE-1)*KMAX+NCKA
      KIEM=KIE-KMAX
      IPT=IPTC(IEA)
      IF(IPT.LE.0)GO TO 9613
      DO 9618 KC=1,IPT
      KA=KC+KIE
      KB=KC+KIEM
      NPC=IPCNT(KA)
      TCVM=TC(KA)+1
      NPL=0
      IF(IE.GT.1)NPL=IPCNT(KB)
      IF(NPC.LE.NPL)GO TO 9618
      NPL=NPL+1
      DO 9619 IPE=NPL,NPC
      IADDR=IPE+(KC-1)*IMXJMX+NCADR
      IF(I.NE.IPC3(IADDR)) GO TO 9619
      IF(IACT(I).LT.-NIDP)GO TO 9614
      INDX=TATR(I)-TCVM
      IEQL=I+(INDX+1)*LMDP
      JEQL=I+(INDX*LM+1)*NIDP
      IF(TATR(IEQL).NE.0.)GO TO 9619
      IEQL=TATR(JEQL)
      IF(IEQL.NE.-IACT(I))GO TO 9614
      IPC3(IADDR)=-IACT(I)
      GO TO 9619
9614  IPC3(IADDR)=0
9619  CONTINUE
9618  CONTINUE
9613  CONTINUE
      IF(IACT(I).GE.-NIDP)GO TO 9517
      IACT(I)=0
```

```
      TATR(I)=0.
      GO TO 9512
 9517 DO 9513 J=1,LDB
      KEQL=I+(LM*(J-1)+1)*NIDP
      JEQL=TATR(KEQL)
      IEQL=I+J*LMDP
      IEQL=TATR(IEQL)
      IF(JEQL.EQ.-IACT(I).AND.IEQL.EQ.0)
     +   GO TO 9514
 9513 CONTINUE
      GO TO 9611
 9514 TATR(KEQL)=0.
 9611 DO 9612 K=2,LDB
      IEQL=I+K*LMDP
      JEQL=I+((K-1)*LM+1)*NIDP
      IF(TATR(IEQL).NE.0..AND.TATR(JEQL).EQ.0.)
     +   TATR(IEQL)=0.
 9612 CONTINUE
      IACT(I)=1
 9512 CONTINUE
      IF(NA.EQ.1)GO TO 1030
C
C                END OF ASSOCIATION LOOPS
C
 952  DO 991 I=1,NMX
      IA=I+(LDX-1)*NIDP
      IF(IACT(I).EQ.0)GO TO 991
 961  IF(TATR(IA).EQ.0.)GO TO 9912
      IEQL=TATR(IA)
      IF(IEQL.LE.0)IEQL=-IEQL
      IF(IEQL.EQ.NAX-1)GO TO 971
      GO TO 991
C
C      CHECK BACKROUND COMING DOWN
C
 971  INBR=0
      ITERM=1
      DO 9716 J=1,LDBM
      IEQL=I+((J-1)*LM+1)*NIDP
      JEQL=IEQL+NIDP
      IF(TATR(IEQL).LE.0..OR.TATR(JEQL).EQ.0.)GO TO 9982
 9716 CONTINUE
      NMP=I+NPDP
      IEQL=TATR(NMP)
      J=1
      IF(JEM.LT.2)GO TO 968
      DO 9711 J=1,JEM
  968 JA=J+NEMB
      IF(IEQL.NE.J)GO TO 9711
 9712 IPB=IPTC(JA)
      DO 9713 K=1,IPB
      KA=(J-1)*KMAX+NBKA
      KAP=K+KA
      KAM=K+KA-KMAX
      IEQL=TATR(I)
      IEQL=IEQL-TC(KAP)
      IF(IEQL.NE.LDB)GO TO 9713
      NP=IPCNT(KAP)
      NL=0
      IF(J.GT.1)NL=IPCNT(KAM)
      NL=NL+1
      DO 9713 N=NL,NP
      IEQL=N+(K-1)*IMXJMX+NBADR
      IF(I.NE.IPC3(IEQL))GO TO 9713
      INBR=INBR+1
      ITERM=2
```

```
      IADDR=N+(K-1)*IMXJMX+NBHDR
      IST=IPC1(IADDR)
      ISP=IPC2(IADDR)+1
      DO 9715 L=IST, ISP
      IU=W(L)
      IF(IU. EQ. -999)GO TO 9715
      IF(IABS(IU). GT. TC(KAP))GO TO 9982
 9715 CONTINUE
 9713 CONTINUE
 9711 CONTINUE
      ITERM=3
      IF(INBR. EQ. 0)GO TO 9982
      ITERM=4
      IID=I+ID2*NIDP
      IF(TATR(IID). LE. TATRMN)GO TO 9982
      DO 981 J=1, LMM
      IEQL=I+(ID2+J-1)*NIDP
  981 UP(J)=TATR(IEQL)
      UP(2)=UP(2)/UP(1)
      UP(6)=A*UP(2)+UP(5)
      DO 985 M=1, NOFST
      MG=6+(M-1)*KOFST+NAN1
      IF(UP(6). GT. ECL(MG))GO TO 986
  985 CONTINUE
      GO TO 989
  986 NAN=NAN+1
      IF(NAN. GT. 1)NAN=0
      NAB=NAN1
      NAN1=NAN*ICLAD
      LMT=NOFST-1
      DO 988 J=1, LMT
      KJ=(J-1)*KOFST
      DO 987 K=1, LM
      JK=K+KJ
      JK1=JK+NAB
      JK2=JK+NAN1
      IF(J. EQ. M) ECL(JK2)=UP(K)
      IF(J. GE. M) JK2=JK2+KOFST
  987 ECL(JK2)=ECL(JK1)
  988 CONTINUE
  989 CONTINUE
      NCELL=NCELL+1
      ITERM=5
      GO TO 9982
 9912 DO 9913 J=1, LDB
      INDP1=I+J*LMDP
      IEQL=TATR(INDP1)
      IF(IEQL. LT. 0)IEQL=-IEQL
      IF(IEQL. EQ. NA)GO TO 991
      IF(TATR(INDP1). LT. 0. . AND. TATR(INDP1). NE. -999. )GO TO 991
 9913 CONTINUE
      ITERM=7
 9982 CONTINUE
      IF(I. LT. NIMN) NIMN=I
      DO 982 J=1, NUMP
      JA=I+(J-1)*NIDP
  982 TATR(JA)=0.
      IACT(I)=0
  991 CONTINUE
 1030 CONTINUE
 1031 CONTINUE
C037  FORMAT(3I15)
 1040 DO 1 J=1, IEM
      JA=J+NEMC
      ISTART=IC1(JA)
      ISTOP=IC2(JA)
```

```
      DO 2 I=ISTART,ISTOP
      MH=W(I-1)
      IF(W(I).GT.MH)MH=W(I)
      IF(W(I+1).GT.MH)MH=W(I+1)
    2 HB(I)=MH
    1 CONTINUE
      RETURN
      END
```

```
*
07/19/79  12:59:56
***LISTING FOR ERT1:TRACK.FTN
$N
      SUBROUTINE TRACK
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 T,TA
      COMMON /INTL/ MHSN,MNSN,HM,FNSN
      COMMON /NVLIS/ NVARM,NCARM,NVO,ICO,IO,JO,JYR,KTL
      COMMON /TLIS/ T,TA,JDAY,JHR,JMIN,JSEC,IDAY,IHR,MIN,ISEC
      COMMON /PNTRS/ NCMX,NVMIN,NVMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /DATA1/ ECL(224),NOFST,KOFST,ICLAD,NAN1
      COMMON /ECONST/ EARTH,VMK
      COMMON /CNT/ COSPHI,SINPHI,COSPH2
      COMMON /AZ2/ SINA,COSA,DELTAZ,ISCANF,NEL,RI,SA
      COMMON /KTA/NV,NC,UCN
C
C          INITIALIZE
C
      UCN=0.
      VKM=COSPHI*VMK
      VKM2=VKM*VMK
      SAVKM2=SA*VKM2
      SAVKM=SA*VKM
      NAN2=NAN1+1
      DO 10 M=1,NOFST
      MA=(NOFST-M)*KOFST+NAN2
      IF(ECL(MA).GT.0.) GO TO 22
   10 CONTINUE
      GO TO 41
   22 NCMX=NOFST-M+1
      DO 30 M=1,NCMX
      M1=1+(M-1)*KOFST+NAN1
      M2=1+M1
      M3=1+M2
      M4=1+M3
      M5=1+M4
      M6=1+M5
      M7=1+M6
C     WRITE(6,50)M1,M2,M3,M4,M5,M6,COSPHI,VKM,ECL(M1),ECL(M2),
C    +     ECL(M3),ECL(M4),ECL(M5),ECL(M6),NAN1
C  50 FORMAT(1X,6I6,4X,3HCOS,F10.5,4X,3HVKM,F10.2,/4HECL-,
C    +     6F8.2,4X,4HNAN1,I6)
      ECL(M5)=ECL(M5)*SAVKM
      VKME=VKM/(ECL(M2)*ECL(M1))
      ECL(M3)=ECL(M3)*VKME
      ECL(M4)=ECL(M4)*VKME
      ECL(M1)=ECL(M1)*SAVKM2
      R2=ECL(M3)*ECL(M3)+ECL(M4)*ECL(M4)
      ECL(M7)=ECL(M6)
   30 ECL(M6)=(SQRT(R2))*SINPHI+R2*COSPH2
      FNSN=FNSN+1.
      NSCAN=NSCAN+1
      IF(NSCAN.NE.1)GO TO 41
      KTL=T
      JDAY=IDAY
      JHR=IHR
      JMIN=MIN
      JSEC=ISEC
      DO 40 NC=1,NCMX
      NV=NC
      CALL ATRAK
   40 CALL BTRAK
```

```
      NVMX=NCMX
      NVMIN=NVMX
      GO TO 45
   41 CONTINUE
      IF(IELSN .LT. IESNL) CALL STRAK
      IESNL=IELSN
      CALL COMPAR
   45 CONTINUE
      RETURN
      END
*
```

```
*
07/19/79  13:00:41
***LISTING FOR ERT1:ATRAK.FTN
$N
      SUBROUTINE ATRAK
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 T,TA
      LOGICAL PR1,PR2,PRIN2
      INTEGER*4 IVCL(736)
      COMMON /ZLOOK/ IZOFF,ZARY(91)
      COMMON /TLIS/ T,TA,JDAY,JHR,JMIN,JSEC,IDAY,IHR,MIN,ISEC
      COMMON /NVLIS/ NVARM,NCARM,NVO,ICO,IO,JO,JYR,KTL
      COMMON /PNTRS/ NCMX,NVMIN,NVMX,IELSN,NSCAN,IESNL,NVSCN,NTT
      COMMON /INTL/ MHSN,MNSN,HM,FNSN
      COMMON /FLGS/ ITYPE,PR1,PR2,PRIN2
      COMMON /DVAL/ DELA
      COMMON /DATA2/ VCL(736),MXVC,NVC1
      COMMON /DATA1/ ECL(224),NOFST,KOFST,ICLAD,NAN1
      COMMON /KTA/NV,NC,DELW
      EQUIVALENCE(VCL(1),IVCL(1))
      IF(NC.LE.0.OR.NC.GT.NCMX) GO TO 100
      IF(NV.LE.0.OR.NV.GT.NVARM) GO TO 100
      NCEC=(NC-1)*KOFST+NAN1
      NVVC=(NV-1)*NVC1
      NCA=2+NCEC
      IZ=ECL(NCA)
      NCA=NCA+1
      X=ECL(NCA)
      NCA=NCA+1
      Y=ECL(NCA)
      NCA=6+NCEC
      H=ECL(NCA)
      NVA=9+NVVC
      IF(IVCL(NVA) .GT. 0)  GO TO 10
C
C     DEFINE LOWEST ELEVATION VALUES
C
      NVA=1+NVVC
      VCL(NVA)=X
      NVA=NVA+1
      VCL(NVA)=Y
      NVA=NVA+1
      IVCL(NVA)=IZ
      NVA=NVA+1
      NCA=NCEC+1
      VCL(NVA)=ECL(NCA)
      NVA=NVA+1
      IVCL(NVA)=T
      NVA=NVA+1
      VCL(NVA)=H
      NCA=3+NCEC
      NCA1=4+NCEC
      R2=ECL(NCA)*ECL(NCA)+ECL(NCA1)*ECL(NCA1)
      R=SQRT(R2)
      NVA=NVA+1
      VCL(NVA)=R
      NVA=NVA+1
      IF(IVCL(NVA) .NE. 0) GO TO 10
      NTT=NTT+1
      IVCL(NVA)=NTT
C
C     INCREMENT ATTRIBUTE ARRAYS
C
```

```
   10 IZL=IZ+IZOFF
      IF(IZL.LT.1) IZL=1
      IF(IZL.GT.91) IZL=91
      Z=ZARY(IZL)
   17 NVA=9+NVVC
      IVCL(NVA)=IVCL(NVA)+1
      NCA=7+NCEC
      NVA=23+NVVC
      VCL(NVA)=VCL(NVA)+ECL(NCA)
      NVA=10+NVVC
      VCL(NVA)=VCL(NVA) + Z
      NVA=NVA+1
      VCL(NVA)=VCL(NVA) + Z*X
      NVA=NVA+1
      VCL(NVA)=VCL(NVA) + Z*Y
      NVA=14+NVVC
      HL=VCL(NVA)
      IF(HL .GT. 0. .OR. FNSN .LT. 2.) GO TO 11
      HL=H-DELA*R
   11 NCA=NCEC+1
      NVA=13+NVVC
      VCL(NVA)=VCL(NVA)+Z*(H-HL)*ECL(NCA)
C
C      SUMMIT VALUES
C
      NVA=NVA+1
      VCL(NVA)=H
      NVA=NVA+1
      IVCL(NVA)=IZ
C
C     PEAK,BASE,AND TOP ARRAYS
C
      NVA=NVA+1
      IZP=IVCL(NVA)
      IF(IZP.GT.IZ) GO TO 99
C
C     SET PEAK
C
   32 NVA=16+NVVC
      IVCL(NVA)=IZ
      NVA=NVA+1
      VCL(NVA)=H
   99 IF(.NOT.PRIN2) GO TO 100
      NCA=NCEC+1
      NVA=9+NVVC
  100 RETURN
      END
*
```

```
*
07/19/79  13:01:37
***LISTING FOR ERT1:BTRAK.FTN
$N
      SUBROUTINE BTRAK
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 T,TA
      INTEGER *4 IVR(192)
      COMMON /DATA1/ ECL(224),NOFST,KOFST,ICLAD,NAN1
      COMMON /DATA3/ VR(192),MXVR,NVR1
      COMMON /TLIS/ T,TA,JDAY,JHR,JMIN,JSEC,IDAY,IHR,MIN,ISEC
      COMMON /NVLIS/ NVARM,NCARM,NVO,ICO,IO,JO,JYR,KTL
      COMMON /KTA/ NV,NC,UCN
      EQUIVALENCE(VR(1),IVR(1))
      IF(NV.LE.0.OR.NV.GT.NVARM) GO TO 10
      IF(NC.LE.0.OR.NC.GT.NCARM) GO TO 10
C
C     DEFINE LAST ELEVATION VALUES
C
      NCEC=(NC-1)*KOFST+NAN1
      NVVR=(NV-1)*NVR1
      NCA=3+NCEC
      NVA=1+NVVR
      VR(NVA)=ECL(NCA)
      NCA=NCA+1
      NVA=NVA+1
      VR(NVA)=ECL(NCA)
      NCA=2+NCEC
      NVA=NVA+1
      VR(NVA)=ECL(NCA)
      NCA=NCA-1
      NVA=NVA+1
      VR(NVA)=ECL(NCA)
      NVA=NVA+1
      IVR(NVA)=T
      NVA=NVA+1
      NCA=6+NCEC
      VR(NVA)=ECL(NCA)
   10 RETURN
      END
*
```

```
*
07/19/79  13:02:10
***LISTING FOR ERT1:COMPAR.FTN
$N
      SUBROUTINE COMPAR
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 T,TA
      LOGICAL PR1,PR2,PRIN2,ARRAY
      INTEGER*4 IVCL(736),IVR(192)
      COMMON /DATA1/ ECL(224),NOFST,KOFST,ICLAD,NAN1
      COMMON /DATA2/ VCL(736),MXVC,NVC1
      COMMON /DATA3/ VR(192),MXVR,NVR1
      COMMON /CDRAYS/ IC(32,10),C(32,9),ID(32,10),D(32,9),IM,JM
      COMMON /FLGS/ ITYPE,PR1,PR2,PRIN2
      COMMON /VPARM/ VX,VY
      COMMON /PNTRS/ NCMX,NVMIN,NVMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /CONST/ VMISW,DIV,VMAG,VMISWM,ZDIV,ADIV,A1,A2,A3,B1,B2,HDIV
      COMMON /TLIS/ T,TA,JDAY,JHR,JMIN,JSEC,IDAY,IHR,MIN,ISEC
      COMMON /NVLIS/ NVARM,NCARM,NVO,ICO,IO,JO,JYR,KTL
      COMMON /CNT/ COSPHI,SINEL,COSPI2
      COMMON /INTL/ MHSN,MNSN,HM,FNSN
      COMMON /KTA/ NV,NC,UCN
      COMMON /RSLV/ IUV1(32),IUV2(32),IUC1(16),IUC2(16),
     +    UV(32),UC(16),NCR
      EQUIVALENCE(VCL(1),IVCL(1)),(VR(1),IVR(1))
      IDV=1
      ICV=1
      IF(NCMX.LE.0) RETURN
      NVMXP=1
      IF(NVMX.GT.1) NVMXP=NVMX
      DO 3 I=1,NVMXP
      UV(I)=0.
      IUV1(I)=0
    3 IUV2(I)=0
      DO 4 I=1,NCMX
      UC(I)=0.
      IUC1(I)=0
    4 IUC2(I)=0
      DO 5 I=1,IM
      IC(I,1)=0
      ID(I,1)=0
      DO 7 J=1,JM
      J1=J+1
      IC(I,J1)=0
      ID(I,J1)=0
      C(I,J)=0.
      D(I,J)=0.
    7 CONTINUE
    5 CONTINUE
      DO 10 NC=1,NCMX
      NCEC=(NC-1)*KOFST+NAN1
      NC1=1+NCEC
      NC2=NC1+1
      NC3=NC2+1
      NC4=NC3+1
      NC6=NC4+2
      NVC=0
      DO 40 NV=1,NVMXP
      NVVC=(NV-1)*NVC1
      NVVR=(NV-1)*NVR1
      MLAST=0
      DELWL=0.
      NLR=20+NVVC
```

```
      NLR1=5+NVVC
      NLR2=5+NVVR
      IF(IVCL(NLR) LE 0 . AND. IVCL(NLR1) LE 0) GO TO 40
      DTTA=T-IVR(NLR2)
      ATEST=(VMAG*DTTA)*(VMAG*DTTA)+VMISWM
      NRA=1+NVVR
      NCA=21+NVVC
      DELX=ECL(NC3)-VR(NRA)-VCL(NCA)*DTTA
      DELX2=DELX*DELX
      IF(DELX2 . GT. ATEST) GO TO 20
      NRA=2+NVVR
      NCA=21+NVVC
      DELY=ECL(NC4)-VR(NRA)-VCL(NCA)*DTTA
      DELY2=DELY*DELY
      IF(DELY2 . GT. ATEST) GO TO 20
C
C          ASSOCIATED, FIND BEST
C
      NRA=3+NVVR
      NRA1=NRA+1
      NRA2=NRA1+2
      DELW=ABS(ECL(NC2)-VR(NRA))*ZDIV+(DELX2+DELY2)*
     1     DIV+1.  + ABS(ECL(NC1)-VR(NRA1))*ADIV
     2  +ABS(ECL(NC6)-VR(NRA2))*HDIV
      IF(DELWL. GT. VMISW) MLAST=1
      GO TO 20
   30 NVC=NVC+1
      IF(NVC . GT.  1) GO TO 31
      IUC1(NC)=NV
      UC(NC)=DELW
      GO TO 33
   31 IF(IUC1(NC) . EQ.  NV) GO TO 33
      NVT=NV
      IF(DELW . LT.  UC(NC)) GO TO 32
   35 IF(IUC2(NC) . LE.  0) GO TO 36
      I=IUC2(NC)
      IF(I. GT. IM) GO TO 361
   39 J=ID(I,1)+1
      ID(I,1)=J
      IF(J . LE.  JM) GO TO 37
C     PRINT 101, JM,NSCAN,NVT,NC,J
      JO=JO+1
      J=JM
   37 ID(I,J+1)=NVT
      D(I,J)=DELW
      GO TO 33
   36 I=IDV
      IDV=I+1
      IF(I . LE.  IM) GO TO 38
C     PRINT 102, IM,NSCAN,NVT,NC,I
  361 IO=IO+1
      I=IM
   38 IUC2(NC)=I
      GO TO 39
   32 DX=UC(NC)
      UC(NC)=DELW
      DELW=DX
      NVT=IUC1(NC)
      IUC1(NC)=NV
      GO TO 35
   33 IF(IUV1(NV) . NE.  0) GO TO 21
      IUV1(NV)=NC
      UV(NV)=DELW
      GO TO 40
C
C          CLUSTER
```

```
C
   21 IF(IUV1(NV) .EQ. NC) GO TO 40
      NCT=NC
      IF(DELW .LT. UV(NV)) GO TO 22
   25 IF(IUV2(NV) .EQ. 0) GO TO 26
      I=IUV2(NV)
      IF(I GT IM) GO TO 261
   29 J=IC(I,1)+1
      IC(I,1)=J
      IF(J .LE. JM) GO TO 27
C     PRINT 101,JM,NSCAN,NV,NCT,J
C 101 FORMAT(' NO. OF CELLS IN CLUSTER EXCEEDS JM = ',5I10)
      JO=JO+1
      J=JM
   27 J1=J+1
      IC(I,J1)=NCT
      C(I,J)=DELW
      GO TO 40
   26 I=ICV
      ICV=I+1
      IF(I .LE. IM) GO TO 28
C     PRINT 102,IM,NSCAN,NV,NCT,I
C 102 FORMAT(' NO. OF ENTRIES IN CLUSTER ARRAY EXCEEDS IM =',
C    1       5I10)
  261 IO=IO+1
      I=IM
   28 IUV2(NV)=I
      GO TO 29
   22 DX=UV(NV)
      UV(NV)=DELW
      DELW=DX
      NCT=IUV1(NV)
      IUV1(NV)=NC
      GO TO 25
C
C          NO COMPAR V, TRY VCL
C
   20 NCA=5+NVVC
      DELT=T-IVCL(NCA)
      ATEST=VMISWM+(VMAG*DELT)*(VMAG*DELT)
      NCA=1+NVVC
      NCA1=21+NVVC
      DELX=ECL(NC3)-VCL(NCA)-VCL(NCA1)*DELT
      DELX2=DELX*DELX
      IF(DELX2 .GT. ATEST) GO TO 40
      NCA=2+NVVC
      NCA1=22+NVVC
      DELY=ECL(NC4)-VCL(NCA)-VCL(NCA1)*DELT
      DELY2=DELY*DELY
      IF(DELY2 .GT. ATEST) GO TO 40
      NCA=3+NVVC
      NCA1=NCA+1
      NCA2=NCA1+2
      DELW=ABS((ECL(NC2)-FLOAT(IVCL(NCA)))*ZDIV)+((DELX2
     1     +DELY2)*DIV)+1. + ABS(ECL(NC1)-VCL(NCA1))*ADIV
     2   +ABS(ECL(NC6)-VCL(NCA2))*HDIV
      IF(MLAST.NE.0.AND.DELWL.LT.DELW)DELW=DELWL
      IF(DELW.LE.VMISW) GO TO 30
   40 CONTINUE
      IF(NVC .GT. 0) GO TO 10
C
C          ISOLATED CELL, NO COMPAR
C
      IF(NVMX.LT.NVMIN) GO TO 501
      DO 50 NV=NVMIN,NVMX
      NVVC=(NV-1)*NVC1
```

```
      NCA=20+NVVC
      NCA1=9+NVVC
      IF(IVCL(NCA).EQ.0 .AND. IVCL(NCA1).EQ.0) GO TO 55
   50 CONTINUE
  501 NV=NVMX+1
      IF(NV .LE. NVARM) GO TO 51
C     PRINT 103,NVARM,NV
C 103 FORMAT(' (C) NO. OF ACTIVE TRACKS EXCEEDS ARRAY MAXIMUM NVMX =',
C    1       2I10)
      NVO=NVO+1
      NV=NVARM
   51 NVMX=NV
   55 NVMIN=NV
      IF(NVMX.LE.0) NVMX=NV
      UCN=UC(NC)
      CALL ATRAK
      CALL BTRAK
      IUV1(NV)=-NC
      IUC1(NC)=-NV
      UV(NV)=0.0
      UC(NC)=0.0
   10 CONTINUE
C
C          HAVE LIST OF COMPARISONS, NOW RESOLV CONFLICTS
C
      DO 60 NV=1,NVMXP
      IF(IUV1(NV) .LT. 0) GO TO 60
      IF(IUV1(NV).EQ.0) GO TO 61
      NC=IUV1(NV)
      IF(NC.GT.NCMX) GO TO 61
      IF(IUC1(NC) .LE. 0) GO TO 61
      IF(IUV2(NV).EQ.0 .AND. IUC2(NC).EQ.0) GO TO 70
      NCR=NC
      CALL RESOLV
      NC=NCR
      GO TO 60
   70 UCN=UC(NC)
      CALL ATRAK
      CALL BTRAK
      IUV1(NV)=-IUV1(NV)
      IUC1(NC)=-IUC1(NC)
      UV(NV)=0.0
      UC(NC)=0.0
      GO TO 60
C    NO NC COMPAR,FIX HEIGHT STATISTICS
   61 IF(FNSN.LT.1.1) GO TO 60
      NVCA=7+(NV-1)*NVC1
      CA=VCL(NVCA)
      HTC=CA*SINEL+CA*CA*COSPI2
   60 CONTINUE
      RETURN
      END
*
```

```
*
07/19/79  13:04:00
***LISTING FOR ERT1:RESOLV.FTN
$N
      SUBROUTINE RESOLV
      LOGICAL PRIN2,PR1,PR2
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 IVCL(736),NVT,IVS,IVT,KV,J,IDIJ,JJ,NCT,ICS,ICT,
     +KC,ICIJ
      DIMENSION V(384)
      COMMON /NVLIS/ NVARM,NCARM,NVO,ICO,IO,JO,JYR,KTL
      COMMON /COMB/ IV(32,7),IVMX
      COMMON /PNTRS/ NCMX,NVMIN,NVMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /CDRAYS/ IC(32,10),C(32,9),ID(32,10),D(32,9),IM,JM
      COMMON /FLGS/ ITYPE,PR1,PR2,PRIN2
      COMMON /DATA2/ VCL(736),MXVC,NVC1
      COMMON /CONST/ VMISW,DIV,VMAG,VMISWM,ZDIV,ADIV,A1,A2,A3,B1,B2,HDIV
      COMMON /CNT/ COSPHI,SINEL,COSPI2
      COMMON /INTL/ MHSN,MNSN,HM,FNSN
      COMMON /KTA/ NV,NC,UCN
      COMMON /RSLV/ IUV1(32),IUV2(32),IUC1(16),IUC2(16),
     +    UV(32),UC(16),NCA
      EQUIVALENCE(VCL(1),IVCL(1))
      IVMX=NVARM
C
C
C          HAVE CLUSTER, ORDER LISTS
C
      DO 4 I=1,6
    4 IV(1,I)=0
      IVT=4
      IVS=5
      ICT=1
      ICS=2
      JV=1
      JC=1
      LV=0
      LC=0
      KV=1
      KC=1
      NCT=NCA
      KOF2=128
      KOF3=256
      IF(NCT.LE.0.OR.NCT.GT.NCMX) GO TO 100
C     PROCESS NCT
   65 IF(IUC1(NCT) .LE.  0.OR.IUC1(NCT).GT.NVMX) GO TO 66
      IF(UC(NCT).LE.0.)GO TO 66
      NVT=IUC1(NCT)
      IF(UC(NCT).GT.0.)UC(NCT)=-UC(NCT)
      CALL COMBIN(NVT,IVS,IVT,KV,J)
      IF(IUC2(NCT) .LE.  0.OR.IUC2(NCT).GT.IM) GO TO 62
      I=IUC2(NCT)
      JX=ID(I,1)
      IF(JX .LE.  0) GO TO 62
      IF(JX.GT.JM) JX=JM
      ID(I,1)=-ID(I,1)
      JX1=JX+1
      DO 611 J=2,JX1
      IDIJ=ID(I,J)
      CALL COMBIN(IDIJ,IVS,IVT,KV,JJ)
  611 CONTINUE
C     PROCESS NVT
   62 IF(IUV1(NVT) .LE.  0.OR.IUV1(NVT).GT.NCMX) GO TO 63
```

```
      IF(UV(NVT).LE.0.)GO TO 63
      NCT=IUV1(NVT)
      IF(UV(NVT).GT.0.)UV(NVT)=-UV(NVT)
      CALL COMBIN(NCT,ICS,ICT,KC,J)
      IF(IUV2(NVT) .LE.  0.OR.IUV2(NVT).GT.IM) GO TO 63
      I=IUV2(NVT)
      JX=IC(I,1)
      IF(JX .LE.  0) GO TO 63
      IF(JX.GT.JM) JX=JM
      IC(I,1)=-IC(I,1)
      JX1=JX+1
      DO 621 J=2,JX1
      ICIJ=IC(I,J)
      CALL COMBIN(ICIJ,ICS,ICT,KC,JJ)
  621 CONTINUE
C
C          RUN COMPAR LIST TO FLUSH OUT FULL SET
C
   63 DO 631 K=JV,KV
      NVT=IV(K,IVS)
      IF(NVT.LE.0.OR.NVT.GT.NVARM) GO TO 631
      IF(UV(NVT).LE.0.) GO TO 631
      IF(IUV1(NVT).GT.0.AND.IUV1(NVT).LE.NCMX) GO TO 64
  631 CONTINUE
      GO TO 66
   64 JV=K
      LC=LC+1
      GO TO 62
   66 DO 661 K=JC,KC
      NCT=IV(K,ICS)
      IF(NCT.LE.0.OR.NCT.GT.NCMX) GO TO 661
      IF(UC(NCT).LE.0.)GO TO 661
      IF(IUC1(NCT).GT.0.AND.IUC1(NCT).LE.NVARM) GO TO 67
  661 CONTINUE
      GO TO 68
   67 JC=K
      LV=LV+1
      GO TO 65
   68 IF(LC .EQ.  0) GO TO 69
      LC=0
      JV=1
      JC=1
      LV=0
      GO TO 63
   69 IF(LV .EQ.  0) GO TO 70
      LV=0
      JV=1
      LC=0
      JC=1
      GO TO 66
C
C          HAVE ORDERED LIST,NOW FIND BEST MATCH
C
   70 IF(KC.LE.1 .OR.  KV.LE.1) GO TO 100
      KV=KV-1
      IF(KV.GT.IVMX) GO TO 100
      KC=KC-1
      IF(KC.GT.IVMX) GO TO 100
      IMSM=0
      DO 701 K=1,KV
      NV=IV(K,IVS)
      IF(NV.LE.0.OR.NV.GT.NVMX)GO TO 701
      IF(UV(NV).LT.0.)UV(NV)=-UV(NV)
  701 CONTINUE
      DO 71 K=1,KC
      KA2=K+KOF2
```

```
      KA3=K+KA3
      V(K)=0.
      V(KA2)=0.
      V(KA3)=0.
      IV(K,ICT)=0
      IV(K,IVT)=0
      IV(K,3)=0
      IV(K,6)=0
      IV(K,7)=0
      NC=IV(K,ICS)
      IF(NC.LE.0.OR.NC.GT.NCMX) GO TO 71
      IF(UC(NC).LT.0.)UC(NC)=-UC(NC)
      NV=IUC1(NC)
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 71
      IF(IUV1(NV).LT.0)GO TO 71
      IF(IUV1(NV).NE.NC)GO TO 711
      V(K)=UC(NC)
      UV(NV)=-UV(NV)
      UC(NC)=-UC(NC)
      GO TO 71
  711 IMSM=IMSM+1
   71 CONTINUE
      IF(KV.LE.(KC-IMSM) .OR.  IMSM.EQ.0) GO TO 75
C
C           FIRST ROUND MIN WEIGHT SELECTION
C
      KNV=0
      KNC=0
      DO 72 K=1,KC
      IF(IV(K,3) .NE.  0) GO TO 72
      NC=IV(K,ICS)
      IF(NC.LE.0.OR.NC.GT.NCMX) GO TO 72
      IF(UC(NC).LE.0.1)GO TO 72
      NV=IUC1(NC)
      IF(NV.LE.0.OR.NV.GT.NVMX)GO TO 724
      IF(UV(NV).GT.0.1)GO TO 725
      IV(K,7)=NV
      KNV=KNV+1
  724 I=IUC2(NC)
      IF(I.LE.0.OR.I.GT.IM) GO TO 721
      JX=ID(I,1)
      IF(JX.LT.0)JX=-JX
      IF(JX.LE.0.OR.JX.GT.JM) GO TO 721
      NV=0
      DWT=999.
      DO 723 J=1,JX
      J1=J+1
      NVT=ID(I,J1)
      IF(NVT.LE.0.OR.NVT.GT.NVMX) GO TO 723
      IF(UV(NVT) .LE.  0.1) GO TO 723
      DELW=D(I,J)
      IF(DELW.LE.0.1) GO TO 723
      IF(DELW.LT.DWT)DWT=DELW
      IF(DWT .EQ.  DELW) NV=NVT
  723 CONTINUE
      IF(NV .LE.  0 .OR.NV.GT.NVMX.OR.DWT.GT.VMISW.OR.DWT.LT..1)
     1 GO TO 721
      GO TO 726
  725 DWT=UC(NC)
  726 KA2=K+KOF2
      V(KA2)=DWT
      IF(UV(NV).GT.0.)UV(NV)=-UV(NV)
      IF(UC(NC).GT.0.)UC(NC)=-UC(NC)
      IV(K,ICT)=NV
      GO TO 72
  721 KNC=KNC+1
```

```
      IF(KNC.GT.KC) GO TO 72
      IV(KNC,IVT)=K
   72 CONTINUE
      IF(KNV.LE.0.AND.KNC.LE.0)GO TO 75
      IF(KNC .EQ. 0.OR.KNC.GT.KC) GO TO 80
C
C          CASCADE REORDER OF COMPAR LIST
C
      J=0
  731 J=J+1
      IF(J .GT. KNC) GO TO 80
      K=IV(J,IVT)
      IF(K.LE.0.OR.K.GT.KC) GO TO 731
      NC=IV(K,ICS)
      IF(NC.LE.0.OR.NC.GT.NCMX) GO TO 739
      NV=IUC1(NC)
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 739
      DO 738 L=1,KC
      IF(IV(L,3).EQ.NV)GO TO 7381
      IF(IV(L,ICT).EQ.NV)GO TO 7382
  738 CONTINUE
      GO TO 739
 7381 NCT=IV(L,ICS)
      DELT=VMISW+V(L)
      GO TO 7383
 7382 NCT=IV(L,ICS)
      KA2=L+KOF2
      DELT=VMISW+V(KA2)
 7383 KT=L
      IF(NCT.LE.0.OR.NCT.GT.NCMX)GO TO 739
      IF(IUC1(NCT).LE.0.OR.IUC2(NCT).LE.0)GO TO 739
      DELW=UC(NC)
      IF(DELW.LT.0. )DELW=-DELW
      I=IUC2(NCT)
      IF(I.LT.0)I=-I
      IF(I .LE. 0.OR.I.GT.IM) GO TO 739
      JX=ID(I,1)
      IF(JX.LT.0)JX=-JX
      IF(JX .LE. 0.OR.JX.GT.JM) GO TO 739
      DWT=999.
      NVB=0
      DWT1=999.
      NV1=0
      DO 732 L=1,JX
      IF(D(I,L).LE.0.1) GO TO 732
      IF(D(I,L).LT.DWT)DWT=D(I,L)
      L1=L+1
      NVT1=IABS(ID(I,L1))
      IF(DWT .EQ. D(I,L)) NVB=NVT1
      IF(NVT1.LE.0.OR.NVT1.GT.NVMX) GO TO 732
      IF(UV(NVT1).GT.0.1.AND.D(I,L).LT.DWT1)DWT1=D(I,L)
      IF(DWT1.EQ.D(I,L)) NV1=NVT1
  732 CONTINUE
      IF(NVB.LE.0.OR.NVB.GT.NVMX.OR.DWT.LE..1.OR.DWT.GT.VMISW)
     1   GO TO 739
  734 IF(NV1.LE.0.OR.NV1.GT.NVMX.OR.DWT1.GT.VMISW.OR.DWT1.LE..1)
     1   GO TO 735
      DELW1=DELW+DWT1
      DELW2=DELW+DWT
      IF(DELW1.GT.DELT) GO TO 735
      IV(K,6)=NV
      KA3=K+KOF3
      V(KA3)=DELW
      IV(KT,6)=NVT1
      KA3=KT+KA3
      V(KA3)=DWT1
```

```
      IF(DELW2.GE.DELW1) GO TO 739
      IV(KT,7)=NVB
      GO TO 739
  735 DO 736 I=1,KC
      IF(IV(I,3).EQ.NVB) GO TO 739
      IF(IV(I,ICT).EQ.NVB) GO TO 737
  736 CONTINUE
      GO TO 739
  737 KA2=I+KOF2
      DELT=DELT+V(KA2)
      DELW2=DELW+DWT+VMISW
      IF(DELW2.GT.DELT) GO TO 739
      IV(K,6)=NV
      KA3=K+KOF3
      V(KA3)=DELW
      IV(KT,6)=NVB
      KA3=KT+KOF3
      KA2=I+KOF2
      V(KA3)=DWT
      IV(I,ICT)=0
      V(KA2)=0.
  739 IV(K,7)=0
      KNV=KNV-1
      GO TO 731
C
C      EXCHANGE PAIRS FOR MIN MEASURE
C
   80 IF(KNV.LE.0.OR.KNV.GT.KC) GO TO 75
      DO 801 K=1,KC
      NVB=IV(K,7)
      IF(NVB.LE.0.OR.NVB.GT.NVMX) GO TO 801
      IF(IUV2(NVB).LE.0) GO TO 801
      NC=IV(K,ICS)
      IF(NC.LE.0.OR.NC.GT.NCMX) GO TO 801
      NV=IV(K,2)
      DO 802 L=1,KC
      IF(NVB.EQ.IV(L,ICT).OR.NVB.EQ.IV(L,3)) GO TO 803
  802 CONTINUE
      GO TO 801
  803 NCB=IV(L,ICS)
      I=IUV2(NVB)
      IF(I.LT.0)I=-I
      JX=IC(I,1)
      IF(JX.LE.0.OR.JX.GT.JM) GO TO 801
      DO 807 J=1,JX
      J1=J+1
      IF(IC(I,J1).EQ.NCB) GO TO 808
  807 CONTINUE
      GO TO 801
  808 DSET=C(I,J)
      DUC=UC(NC)
      IF(DUC.LT.0.)DUC=-DUC
      DELWB=DSET+DUC
      KA2=K+KOF2
      KA3=K+KOF3
      DELW1=V(KA3)
      KA=KA-KOFST
      IF(DELW1.LE..1) DELW1=V(KA2)
      IF(DELW1.LE..1) DELW1=V(K)
      KA2=L+KOF2
      KA3=L+KOF3
      DELW2=V(KA3)
      IF(DELW2.LE..1) DELW2=V(KA2)
      IF(DELW2.LE..1) DELW2=V(L)
      DELW=DELW1+DELW2
      IF(DELW.LE.DELWB) GO TO 801
```

```
      KA3=K+KOF3
      V(KA3)=UC(NC)
      IF(V(KA3).LT.0.)V(KA3)=-V(KA3)
      KA3=L+KOF3
      V(KA3)=DSET
      IV(K,6)=NVB
      IV(L,6)=NV
  801 IV(K,7)=0
   75 CONTINUE
C
C     UPDATE ATTRIBUTES
C
      DO 78 K=1,KC
      KA3=K+KOF3
      KA2=K+KOF2
      IF(PRIN2)
     1  WRITE(6,788) IV(K,ICS),IV(K,6),IV(K,ICT),
     2  IV(K,3),V(KA3),V(KA2),V(K)
  788 FORMAT(1X,4I5,3F8.2)
      NC=IV(K,ICS)
      IF(NC.LE.0.OR.NC.GT.NCMX) GO TO 78
      IF(IUC1(NC).LE.0) GO TO 78
      NV=IV(K,6)
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 810
      DWT=V(KA3)
      GO TO 820
  810 NV=IV(K,ICT)
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 811
      DWT=V(KA2)
      GO TO 820
  811 NV=IV(K,3)
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 79
      DWT=V(K)
  820 IF(DWT.LE.0.1.OR.DWT.GT.VMISW) GO TO 79
      IF(IUV1(NV).EQ.0) GO TO 79
      UCN=DWT
      CALL ATRA
      CALL BTRA
      GO TO 77
   79 IF(NVMX.LT.NVMIN) GO TO 7911
      DO 791 I=NVMIN,NVMX
      NVVC=(I-1)*NVC1
      IVA=9+NVVC
      IVA1=IVA+11
      IF(IVCL(IVA1).EQ.0 .AND. IVCL(IVA).EQ.0) GO TO 792
  791 CONTINUE
 7911 I=NVMX+1
      IF(I.LE. NVARM) GO TO 7921
C     PRINT 103,NVARM,I
C 103 FORMAT(' (R) NO. OF ACTIVE TRACKS EXCEEDS ARRAY MAXIMUM NVMX =',
C    1    2I10)
      NVO=NVO+1
      I=NVARM
      GO TO 77
 7921 NVMX=I
  792 NV=I
      NVMIN=I
      UCN=0.
      CALL ATRAK
      CALL BTRAK
   77 IUV1(NV)=-NC
      IUC1(NC)=-NV
      UV(NV)=0.
      UC(NC)=0.
   78 CONTINUE
      DO 99 K=1,NV
```

```
      NV=IV(K, IVS)
      IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO 99
      IF(IUV1(NV). LE. 0) GO TO 99
      IUV1(NV)=-IUV1(NV)
      IF(FNSN. LT. 1. 1) GO TO 99
      NVA=7+(NV-1)*NVC1
      VA=VCL(NVA)
      HTC=VA*SINEL+VA*VA*COSPI2
   99 CONTINUE
  100 RETURN
      END
*
```

```
*
07/19/79  13:07.55
***LISTING FOR ERT1:COMBIN.FTN
$N
      SUBROUTINE COMBIN(N,IS,IT,K,J)
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*4 N,IS,IT,K,J
      COMMON /COMB/ IV(32,7),IVMX
C
C     INSERT N INTO ORDERED ARRAY IV(K,IT)
C     RETURN NEW ARRAY AS IV(K,IS)
C
      I=IS
      IS=IT
      IT=I
      L=0
      DO 10 J=1,K
      IVJ=IV(J,IT)
      IF(IVJ.LT.0)IVJ=-IVJ
      IF(IVJ-N) 20,30,40
   20 IF(IVJ.EQ.0) GO TO 40
   10 IV(J,IS)=IV(J,IT)
      J=K
   40 L=1
      IV(J,IS)=N
   30 DO 50 I=J,K
      IL=I+L
   50 IV(IL,IS)=IV(I,IT)
      K=K+L
      IF(K.GE.IVMX) GO TO 70
      IV(K,IS)=0
      GO TO 80
   70 WRITE(6,100) K,IVMX
  100 FORMAT(' ERROR IN COMBIN,I3,K,IVMX',2X,2I10)
      K=IVMX-1
   80 RETURN
      END
*
```

```
*
07/19/79  13:08:33
***LISTING FOR ERT1:STRAK.FTN
$N
      SUBROUTINE STRAK
      IMPLICIT INTEGER*2 (I-N)
      INTEGER*2 T,TM,TA
      INTEGER*4 IVCL(736),ITCL(21),IVR(192)
      DIMENSION ATST(62),NUM(62)
      LOGICAL PR1,PR2,PRIN2
      DIMENSION TCL(21),DUM(6)
      COMMON /TMAX/ TM
      COMMON /FLGS/ ITYPE,PR1,PR2,PRIN2
      COMMON /PNTRS/ NCMX,NVMIN,NVMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /TLIS/ T,TA,JDAY,JHR,JMIN,JSEC,IDAY,IHR,MIN,ISEC
      COMMON /NVLIS/ NVARM,NCARM,NVO,ICO,IO,JO,JYR,KTL
      COMMON /CONST/ VMISW,DIV,VMAG,VMISWM,ZDIV,ADIV,A1,A2,A3,B1,B2,HDIV
      COMMON /INTL/ MHSN,MNSN,HM,FNSN
      COMMON /VPARM/ VX,VY
      COMMON /DATA1/ ECL(224),NOFST,KOFST,ICLAD,NAN1
      COMMON /DATA2/ VCL(736),MXVC,NVC1
      COMMON /DATA3/ VR(192),MXVR,NVR1
      EQUIVALENCE(TCL(1),ITCL(1))
      EQUIVALENCE(VCL(1),IVCL(1)),(VR(1),IVR(1))
      DATA IZERO/0/,IOUT/31/,LAN/62/
      MAN=0
      NVOUT=NVMX-1
      DO 19 N=1,LAN
      ATST(N)=0.
   19 NUM(N)=0
C
C
      VXC=0
      VYC=0
      NSN=0
      NACT=0
      NFN=FNSN-1.
      IF(NFN.LE.0) GO TO 55
C     WRITE(2) KTL,NVSCN,JDAY,JHR,JMIN,JSEC
C    1 ,NFN,NSCAN,NT,NVMX,JYR,NVO
C    2 ,IO,JO,IZERO,VX,VY,DUM
      IF(PR1.OR.PR2)
     1WRITE(6,1000) JDAY,JHR,JMIN,JSEC,NVSCN
 1000 FORMAT(1H1,3X,'PROGRAM TRACK4 (781207) ',5X,' TRACK TIME',4I4,
     15X,'VOL SCAN',I6//,1X,'TRK   LOCATION DBZ   CELL    PEAK   ',
     2'  SURFACE   SUMMIT   VELOCITY      A'/,1X
     3' ID EAST NORTH AV   VOL DBZ   HT    DBZ   HT   DBZ   HT',
     4' EAST NORTH    L'/,1X,
     5'        KM   KM          KM3        KM          KM         KM',
     6'    M/S   M/S    L')
      DO 20 NV=1,NVOUT
      NVA=(NV-1)*NVC1
      NV23=23+NVA
      A=VCL(NV23)
      DO 21 N=1,IOUT
      NN=N+MAN
      IF(A.GT.ATST(NN)) GO TO 22
   21 CONTINUE
      NN=NN+1
   22 LAN=MAN
      MAN=MAN+IOUT
      IF(MAN.GT.IOUT) MAN=0
      IFML=0
```

```
      DO 23 M=1,IOUT
      MM=M+MAN
      ML=M+LAN
      IF(ML.NE.NN.OR.IFML.EQ.1) GO TO 24
      ATST(MM)=A
      NUM(MM)=NV
      LAN=LAN-1
      IFML=1
      GO TO 23
   24 ATST(MM)=ATST(ML)
      NUM(MM)=NUM(ML)
   23 CONTINUE
   20 CONTINUE
   50 DO 100 N=1,IOUT
      NN=N+MAN
      NV=NUM(NN)
      IF(NV.EQ.0) GO TO 100
      NVA=(NV-1)*NVC1
      NV1=1+NVA
      NV2=NV1+1
      NV3=NV2+1
      NV4=NV3+1
      NV5=NV4+1
      NV6=NV5+1
      NV7=NV6+1
      NV8=NV7+1
      NV9=NV8+1
      NV10=NV9+1
      NV11=NV10+1
      NV12=NV11+1
      NV13=NV12+1
      NV14=NV13+1
      NV15=NV14+1
      NV16=NV15+1
      NV17=NV16+1
      NV18=NV17+1
      NV19=NV18+1
      NV20=NV19+1
      NV21=NV20+1
      NV22=NV21+1
      NV23=NV22+1
      NR1=1+(NV-1)*NVR1
      NR2=NR1+1
      NR3=NR2+1
      NR4=NR3+1
      NR5=NR4+1
      NR6=NR5+1
      IF(IVCL(NV9) .LE. 0) GO TO 200
      ZZDIV=1.0/VCL(NV10)
      ITCL(1)=IVCL(NV5)
      TCL(2)=VCL(NV11)*ZZDIV
      TCL(3)=VCL(NV12)*ZZDIV
      TCL(4)=10.*ALOG10(VCL(NV10)/FLOAT(IVCL(NV9)))
      TCL(5)=VCL(NV13)*ZZDIV
      ITCL(7)=IVCL(NV16)
      TCL(8)=VCL(NV17)
      ITCL(10)=IVCL(NV3)
      TCL(11)=VCL(NV4)
      TCL(12)=VCL(NV6)
      IM3=IVCL(NV16)-3
   30 ITCL(15)=IVCL(NV15)
      TCL(16)=VCL(NV14)
      IVL8=IVCL(NV8)
      IF(IVL8.LT.0)IVL8=-IVL8
      ITCL(17)=IVL8
      ITCL(18)=0
```

```
      ITCL(20)=0
      ITCL(21)=0
      VXT=VCL(NV21)
      VYT=VCL(NV22)
      ITIM=IVCL(NV20)
      JTIM=ITCL(1)
      IF(ITIM.EQ.0 .OR. ITIM.EQ.JTIM) GO TO 40
      DELTM=JTIM-ITIM
      DELTM=1.0/DELTM
      TCL(20)=(TCL(2)-VCL(NV18))*DELTM
      TCL(21)=(TCL(3)-VCL(NV19))*DELTM
      VXT=TCL(20)
      VYT=TCL(21)
   40 IVCL(NV20)=ITCL(1)
      VCL(NV19)=TCL(3)
      VCL(NV18)=TCL(2)
      VCL(NV21)=A1*VXT+A2*VCL(NV21)+A3*VX
      VCL(NV22)=A1*VYT+A2*VCL(NV22)+A3*VY
      VR(NR1)=TCL(2)
      VR(NR2)=TCL(3)
      VR(NR3)=TCL(4)
      VR(NR4)=VCL(NV4)
      IVR(NR5)=ITCL(1)
      VR(NR6)=TCL(8)
      IZVAL=VR(NR3)
      IDTC=FLOAT(IVCL(NV9))/(FNSN-1.)*10.+.5
      NSN=NSN+1
      VXC=VXC+TCL(20)
      VYC=VYC+TCL(21)
      VXP=VCL(NV21)*1000.
      VYP=VCL(NV22)*1000.
      NACT=NACT+1
C     WRITE(2) TCL,VCL(NV39),VCL(NV40),IDTC
      IF(PR1) GO TO 59
      IF(.NOT.PR2) GO TO 58
      IF((IDTC.GT.MNSN.OR.IZVAL.GT.35).AND.IVCL(NV8).GT.0)
     1   IVCL(NV8)=-IVCL(NV8)
      IF(IVCL(NV8).GE.0) GO TO 58
   59 WRITE(6,1005) ITCL(17),VR(NR1),VR(NR2),IZVAL,TCL(5),ITCL(7),
     1TCL(8),ITCL(10),TCL(12),ITCL(15),TCL(16),
     2VXP,VYP,IDTC
 1005 FORMAT(1X,I4,2F5.0,I3,F6.1,I3,F5.1,I5,F5.1,I4,
     +   3F5.1,2X,I2)
   58 DO 41 I=9,14
      IA=I+NVA
   41 IVCL(IA)=0
      IVCL(NV23)=0
      GO TO 100
  200 IF(IVCL(NV20) .LE.  0) GO TO 102
      IF((T-IVCL(NV20)) .LE.  TM) GO TO 100
      DO 101 I=1,20
      IA=I+NVA
  101 IVCL(IA)=0
      IA=23+NVA
      VCL(IA)=0.
      IF(NV.LT.NVMIN)NVMIN=NV
  102 IVCL(NV23)=0
  100 CONTINUE
      IF(NSN .EQ.  0) GO TO 56
      VN=NSN
      VX=B1*VXC/VN+B2*VX
      VY=B1*VYC/VN+B2*VY
      DO 43 JI=1,IOUT
      II=IJ+MAN
      I=NUM(II)
      IF(I.EQ.0) GO TO 43
```

```
      NVA=(I-1)*NVC1+20
      IF(IVCL(NVA).GT.0) GO TO 43
      NA=21+(I-1)*NVC1
      VCL(NA)=VX
      NA=NA+1
      VCL(NA)=VY
   43 CONTINUE
      VXP=VX*1000.
      VYP=VY*1000.
   56 IF(NVSCN.GT.1) GO TO 55
      WRITE(6,1003)
 1003 FORMAT(///1X,' VOL   START TIME   NO EL   LAST     TRACK',
     1 '           OVERFLOW        AV VELOCITY'/,1X,
     2 'SCAN  DAY HHMM SS SCANS  SCAN  TOTAL ACT.',
     3 '    NV   IC    I    J    EAST   NORTH'/)
   55 WRITE(6,1004) NVSCN,JDAY,JHR,JMIN,JSEC,NFN,NSCAN,NT,NACT
     1,NVO,ICO,IO,JO,VXP,VYP
 1004 FORMAT(1X,I5,I4,I3,I2,I3,I5,2I6,I5,I7,3I4,1X,2F6.1)
      NVSCN=NVSCN+1
      NVO=0
      ICO=0
      IO=0
      JO=0
      JDAY=IDAY
      JHR=IHR
      JMIN=MIN
      JSEC=ISEC
      FNSN=1.009
      KTL=T
      RETURN
      END
*
```

APPENDIX F

SAMPLE OUTPUT

SAMPLE OUTPUT FILE

Two types of output are produced following each volume scan:

1. A list of attributes describing each of the 31* most significant cells detected and tracked within the volume scan

   a. ITCL (17) - the ID number of this cell track
   b. VR (1) - cell location to east [(-)west] of radar (km)
   c. VR (2) - cell location to north [(-)south] of radar (km)
   d. IZVAL = VR (3) - average reflectivity of cell (dBZ)
   e. TCL (5) - cell volume (km^3)
   f. ITCL (7) - peak reflectivity of cell (dBZ)
   g. TCL (8) - height of peak reflectivity (km)
   h. ITCL (10) - reflectivity at base of cell (dBZ)
   i. TCL (12) - height of cell base (km)
   j. ITCL (15) - reflectivity at summit of cell (dBZ)
   k. TCL (16) - height of cell summit (km)
   l. VXP = VCL (39)*10^3 - cell velocity toward east [(-)west] (m/s)
   m. VYP = VCL (40)*10^3 - cell velocity toward north [(-)south] (m/s)
   n. IDPC = IVCL (9)/(FNSN-1)*10
      number of scans cell detected/number of scans per volume scan
      percent of elevation scans cell detected at

2. A summary list of statistics for the entire volume scan

   a. NVSCN - volume scan number
   b. JDAY, JHR, JMIN, JSEC - start time of volume scan
   c. NFN = FNSN-1 - number of elevation scans processed in this volume scan
   d. NSCAN - number of last elevation scan in the volume scan
   e. NT - number of cell tracks updated this scan
   f. NACT - number of possible cell tracks stored from current and previous scans
   g. NVO - number of significant cells detected but not outputted
   h. ICO - number of internally paired and clustered cells over the dimensions of CLUST array in subroutine RESOLV
   i. IO - number of detected cells over array dimensions of cluster array n subroutine COMPAR
   j. JO - number of detected cells over array dimensions of IC array in subroutine COMPAR
   k. VXP - an estimate of the average velocity east [(-)west] any new cells will have (m/s), set to 0 on first scan
   l. VYP - an estimate of the average velocity north [(-)south] any new cells will have (m/s), set to 0 on first scan

*see Appendix F - Option to Increase Number of Significant Cells

SAMPLE OUTPUT OF FIRST VOLUME SCAN

PROGRAM TRACK4 (781207)   TRACK TIME 120 16 46 59   VOL SCAN 1

TRK ID	LOCATION EAST KM	LOCATION NORTH KM	DBZ AV	CELL VOL KM3	PEAK DBZ	PEAK HT KM	SURFACE DBZ	SURFACE HT KM	SUMMIT DBZ	SUMMIT HT KM	VELOCITY EAST M/S	VELOCITY NORTH M/S	ALL
17	-90.	40.	45	1.5	45	4.3	45	4.3	45	4.3	0.0	0.0	3
1	-60.	-63.	34	2.7	34	1.8	34	1.8	34	1.8	0.0	0.0	3
2	-96.	37.	41	4.0	41	2.3	41	2.3	41	2.3	0.0	0.0	3
18	-89.	32.	47	2.7	47	4.1	47	4.1	47	4.1	0.0	0.0	3
3	-56.	-67.	42	7.8	42	6.7	42	1.8	42	6.7	0.0	0.0	7
19	-80.	38.	36	5.4	36	6.8	36	3.7	36	6.8	0.0	0.0	7
20	-86.	30.	37	2.6	38	3.9	38	3.9	36	7.0	0.0	0.0	7
5	-94.	34.	46	3.8	47	2.1	47	2.1	46	4.4	0.0	0.0	7
21	-97.	53.	39	1.8	39	4.8	39	4.8	39	4.8	0.0	0.0	3
4	-50.	-66.	36	2.4	36	1.7	36	1.7	36	1.7	0.0	0.0	3
31	-101.	35.	41	1.7	41	8.2	41	8.2	41	8.2	0.0	0.0	3
22	-91.	50.	42	6.3	43	4.6	43	4.6	42	8.0	0.0	0.0	7
30	-93.	37.	46	1.5	46	4.3	46	4.3	46	4.3	0.0	0.0	3
6	-43.	63.	37	2.0	37	1.5	37	1.5	37	1.5	0.0	0.0	3
23	-37.	54.	52	2.5	52	2.7	52	2.7	52	2.7	0.0	0.0	3
7	-4.	25.	50	0.2	50	0.4	50	0.4	50	0.4	0.0	0.0	3
14	-72.	53.	44	3.0	45	3.9	43	1.9	45	3.9	0.0	0.0	7
8	-50.	60.	41	4.3	41	1.6	41	1.6	41	1.6	0.0	0.0	3
24	-75.	83.	36	3.7	36	4.9	36	4.9	36	4.9	0.0	0.0	3
25	-73.	56.	39	6.3	40	10.0	38	4.0	40	10.0	0.0	0.0	7
9	-27.	2.	43	0.4	44	0.5	44	0.5	44	0.5	0.0	0.0	3
10	-14.	25.	41	0.2	41	0.5	41	0.5	41	0.5	0.0	0.0	3
11	-6.	15.	42	0.1	42	0.3	42	0.3	42	0.3	0.0	0.0	3
12	-80.	61.	40	3.8	40	2.2	40	2.2	40	2.2	0.0	0.0	3
13	-17.	10.	39	0.2	39	0.3	39	0.3	39	0.3	0.0	0.0	3
15	-4.	53.	49	0.9	49	1.0	49	1.0	49	1.0	0.0	0.0	3
26	-57.	82.	43	1.5	44	4.3	44	4.3	44	4.3	0.0	0.0	3
16	-6.	1.	41	0.1	41	0.1	41	0.1	41	0.1	0.0	0.0	3
27	-54.	80.	42	1.4	42	4.2	42	4.2	42	4.2	0.0	0.0	3
28	-62.	55.	36	1.0	36	3.5	36	3.5	36	3.5	0.0	0.0	3
29	-52.	-70.	45	1.1	45	3.7	45	3.7	45	3.7	0.0	0.0	3

VOL SCAN	START TIME DAY	START TIME HHMM	START TIME SS	NO EL SCANS	LAST SCAN	TRACK TOTAL	TRACK ACT.	OVERFLOW NV	OVERFLOW IC	OVERFLOW I	OVERFLOW J	AV VELOCITY EAST	AV VELOCITY NORTH
1	120	1646	59	3	4	32	31	25	0	0	0	0.0	0.0

APPENDIX F

Options to Increase:

Number of Significant Cells
Number of Active Tracks

Option: Increase Number of Significant Cells Processed Each Scan

Both the real-time program and the post-mission program are set up to process 16 significant cells each scan. To increase the number of cells the dimensions of several arrays and address offsets must be increased.

In block common DATA1 - for each addtional cell

    increase dimension of ECL by 14
    increase NOFST* by 1
    increase ICLAD* by 7

In block common RSLV - for each additional cell

    increase dimension of: IUC1, IUC2&UC by 1

In block common NVLIS - for each additional cell

    increase NCARM* by 1

In subroutine RESOLV - for each additional cell

    increase dimension of V by 24
    increase KOF2&KOF3 by 8

Note: to process more than 32 cells each scan the number of active tracks must be increased to at least that number (see option to increase active tracks).

*variable set in block DATA

Option: Increase Number of Active Tracks Updated Each Scan and Outputted Each Volume Scan

Both the real-time program and the post-mission program are set up to update and output 31 active tracks at any one scan. To increase the number of tracks, the dimensions of several arrays and address offsets must be increased.

In block common DATA2 - for each additional track

increase dimension of VCL and IVCL by 23
increase MXVC* by 23

In block common DATA3 - for each additional track

increase dimension of VR and IVR by 6
increase MXVR* by 7

In block common RSLV - for each additional track

increase dimension of: IUV1,IUV2&UV by 1

In block common CDRAYS - for each additional track

increase the first dimension of: IC,C,ID&D by 1
increase IM* by 1

In block common NVLIS - for each additional track

increase NVARM* by 1

In block common COMB - for each additional track

increase the first dimension of IV by 1

In subroutine STRAK - for each additional track

increase dimension of ATST & NUM by 2
increase IOUT by 1
increase LAN by 2

*variable set in block DATA